



## *Memoria del proyecto*

*Fecha:* Enero - 2014

*Autor:* Raquel Gracia Rosich  
*Consultor:* Jordi Ceballos Villach

## Índice de contenido

|   |    |
|---|----|
| 1 Descripción del proyecto.....   | 4  |
| 1.1 Introducción.....   | 4  |
| 1.2 Objetivos del proyecto.....   | 4  |
| 2 Alcance del proyecto.....   | 4  |
| 2.1 Requerimientos funcionales.....                                       | 4  |
| 2.1.1 Recetas.....  | 4  |
| 2.1.2 Favoritas.....  | 5  |
| 2.1.3 Compra.....   | 5  |
| 2.1.4 Información.....  | 5  |
| 2.2 Requerimientos técnicos.....  | 5  |
| 2.2.1 BaaS (Backend as a service).....                                    | 5  |
| 2.2.2 Brackets.....   | 5  |
| 2.2.3 Codiqa.....   | 5  |
| 2.2.4 CSS3.....   | 5  |
| 2.2.5 Dia.....  | 6  |
| 2.2.6 GanttProject.....   | 6  |
| 2.2.7 HTML5.....  | 6  |
| 2.2.8 JavaScript.....   | 6  |
| 2.2.9 jQuery Mobile.....  | 6  |
| 2.2.10 OpenOffice 4.0.0.....  | 6  |
| 2.2.11 Phonegap Build.....  | 6  |
| 2.2.12 Photoshop CS6.....   | 6  |
| 3 Organización del proyecto.....  | 6  |
| 3.1 Fases.....  | 6  |
| 3.2 Planificación.....  | 7  |
| 4 Riesgos del proyecto.....   | 8  |
| 5 Análisis funcional.....   | 9  |
| 5.1 Funcionalidades de la aplicación.....                                 | 9  |
| 5.2 Casos de usos.....  | 9  |
| 5.2.1 Diagramas casos de usos.....  | 9  |
| 5.2.2 Descripción caso de uso [CU01] Listado Categorías.....              | 10 |
| 5.2.3 Descripción caso de uso [CU02] Listado Recetas.....                 | 10 |
| 5.2.4 Descripción caso de uso [CU03] Ficha Receta.....                    | 11 |
| 5.2.5 Descripción caso de uso [CU04] Enviar ingredientes al L.Compra..... | 11 |
| 5.2.6 Descripción caso de uso [CU05] Añadir/Borrar Receta Favorita.....   | 12 |
| 5.2.7 Descripción caso de uso [CU06] Listado Recetas Favoritas.....       | 12 |
| 5.2.8 Descripción caso de uso [CU07] Listado Compra.....                  | 12 |
| 5.2.9 Descripción caso de uso [CU08] Vaciar Lista Compra.....             | 13 |
| 5.2.10 Descripción caso de uso [CU09] Añadir Ingrediente .....            | 13 |

---

|  |    |
|--|----|
| 5.2.11 Descripción caso de uso [CU10] Información..... | 14 |
| 6 Diseño técnico.....                                  | 14 |
| 6.1 Arquitectura de la aplicación.....                 | 14 |
| 6.1.1 Capa de presentación (vista).....                | 14 |
| 6.1.2 Capa de negocio (controlador).....               | 15 |
| 6.1.3 Capa de datos (modelo).....                      | 15 |
| 6.1.3.1 Parse como BaaS.....                           | 15 |
| 6.2 Diseño base de datos.....                          | 16 |
| 6.2.1 Parse Data.....                                  | 16 |
| 6.2.2 LocalStorage.....                                | 17 |
| 6.3 Diagramas de clases.....                           | 18 |
| 7 Prototipo.....                                       | 18 |
| 7.1 Pantallas: Home y RecetasPage .....                | 19 |
| 7.2 Pantalla: RecetasFicha.....                        | 19 |
| 7.3 Pantalla: FavoritasPage.....                       | 20 |
| 7.4 Pantalla: FavoritasFicha.....                      | 20 |
| 7.5 Pantalla: ComprasPage y DialogoIngrediente.....    | 21 |
| 7.6 Pantalla: InfoPage.....                            | 21 |
| 7.7 Flujo de navegación.....                           | 22 |
| 8 Implementación y pruebas.....                        | 22 |
| 8.1 Código HTML y JS.....                              | 22 |
| 8.2 PhoneGap Build.....                                | 25 |
| 8.3 Capturas aplicación.....                           | 27 |
| 8.4 Parse.....   | 33 |
| 9 Conclusiones.....                                    | 35 |
| 9.1 Logro de objetivos.....                            | 35 |
| 9.2 Líneas futuras.....                                | 35 |
| 10 Instalación de la aplicación.....                   | 36 |
| 11 Fuentes de información.....                         | 37 |
| 11.1 Bibliografía.....                                 | 37 |
| 11.2 Recursos web.....                                 | 37 |
| 12 Índice de figuras.....                              | 38 |

## 1 Descripción del proyecto

El área del proyecto final de carrera (TFC) elegido es el desarrollo de aplicaciones para dispositivos móviles en HTML5.

El proyecto “**Más que postres**” es una evolución de la propuesta inicial del proyecto que consistía en la realización de una aplicación sobre recetas de platos de cocina clasificados por categorías y con opciones de favoritas y lista de compra.

### 1.1 Introducción

A raíz de la necesidad de obtener los contenidos necesarios para generar una base de datos interesante y de calidad para la aplicación, surgió el trabajo colaborativo con Angels Castelló. Angels publica el blog [MAS QUE POSTRES](#) para disfrutar de la cocina y de la pastelería, además de disponer de una [Fan Page](#) en Facebook.

Angels cedió el uso de su recetas y fotografías en el proyecto y como contraprestación la aplicación sería cedida para su explotación con fines no lucrativos una vez terminado el proyecto.

### 1.2 Objetivos del proyecto

El objetivo del proyecto es desarrollar una aplicación multiplataforma con un único desarrollo que facilite al usuario el acceso a un amplio abanico de recetas con sus ingredientes y método de preparación desde su dispositivo móvil. Adicionalmente, otro objetivo es facilitar al usuario la realización de la compra de los ingredientes necesarios, gracias a tener acceso a la lista de la compra desde su dispositivo móvil cuando esté en el supermercado. El usuario podrá interactuar con la aplicación, permitiéndole modificar la lista de la compra generada.

En el ámbito docente, el principal objetivo es aprender y utilizar nuevas tecnologías para lograr desarrollar la aplicación. Es importante el uso de diferentes técnicas de acceso a la información para obtener un mayor conocimiento durante el desarrollo de esta aplicación web móvil híbrida. Por ello se ha decidido que no sea necesario el acceso remoto para la lista de la compra y por otro lado, acceder a través de un servidor remoto a la base de datos de las recetas.

## 2 Alcance del proyecto

### 2.1 Requerimientos funcionales

Para llevar a cabo los objetivos de la aplicación se ha diseñado el siguiente menú de navegación que será el esqueleto de la implementación de las funcionalidades requeridas.

#### 2.1.1 **Recetas**

- Listado de las diferentes categorías de las recetas

- Listado de todas las recetas pertenecientes a una categoría seleccionada
- Consulta ficha receta
  - Marcar/desmarcar receta como favorita
  - Enviar todos los ingredientes a la lista de la compra

### 2.1.2 Favoritas

- Listado recetas favoritas
  - Consultar ficha receta (y sus funcionalidades internas antes mencionadas)

### 2.1.3 Compra

- Listado ingredientes a comprar
  - Borrar ingrediente
  - Añadir manualmente ingrediente

### 2.1.4 Información

- Ficha personal autora recetas
- Enlaces al blog y *Fan Page Facebook*

## 2.2 Requerimientos técnicos

En el proyecto se utilizarán las herramientas y tecnologías que a continuación se detallan en orden alfabético:

### 2.2.1 BaaS (Backend as a service)

Se utilizará un servicio básico para que sea gratuito como *backend*. Entre los posibles *BaaS* a utilizar están *Appcelerator* y *Parse*. En la próxima fase del proyecto, tras su estudio, se decidirá cual de ellos se adapta mejor a las necesidades de la aplicación.

### 2.2.2 Brackets

*Adobe Brackets* es un editor *open source* de código para el desarrollo web escrito en *HTML*, *CSS* y *Javascript* bajo licencia *MIT*. Recientemente han lanzado la versión *Sprint 31*.

### 2.2.3 Codiqa

Es una herramienta *online* para prototipado de aplicaciones móviles multiplataforma.

### 2.2.4 CSS3

Es la tercera especificación oficial de *CSS* (*Cascading Style Sheets*). Hace referencia a un lenguaje de hojas de estilo en cascada que permite dar formato a un documento *HTML*.



### 2.2.5 Dia

Es un programa para dibujar diagramas estructurados de *The Free Software Foundation*.

### 2.2.6 GanttProject

Herramienta gratuita de gestión de proyectos que permite realizar diagramas de GANTT para planificar un proyecto y gestionar los recursos.

### 2.2.7 HTML5

Es la quinta versión del lenguaje de marcado *HTML (Hypertext Markup Language)*.

### 2.2.8 JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar *ECMAScript*, que se utiliza principalmente en su forma del lado del cliente. Actualmente existe una tendencia a su uso también en el lado del servidor.

### 2.2.9 jQuery Mobile

Es un *framework* de desarrollo UI basado en el *core* de *jQuery* que facilita desarrollar aplicaciones web para móviles y es compatible con la mayoría de plataformas.

### 2.2.10 OpenOffice 4.0.0.

Herramienta ofimática libre y gratuita de *Apache Software Foundation*.

### 2.2.11 Phonegap Build

*Phonegap* (también conocido como *Apache Cordova*) es un *framework open source* y gratuito para crear aplicaciones multi-plataforma para dispositivos móviles utilizando *HTML5*, *CSS3* y *JavaScript*.

*PhoneGap Build* es un servicio en la nube que construye rápidamente una aplicación móvil, evitando al programador el trabajo de compilar en diferentes SDKs.

### 2.2.12 Photoshop CS6

Aplicación de Adobe que permite la edición, retoque y optimización de fotografías.

## 3 Organización del proyecto

### 3.1 Fases

Este proyecto se divide en cuatro fases que se corresponden con las cuatro entregas marcadas por el plan docente. Entre la PAC2 y PAC3 hay una entrega extraordinaria que corresponde a la implementación beta del proyecto:



| Nombre                    | Fecha de inicio | Fecha de fin | Duración |
|---------------------------|-----------------|--------------|----------|
| ▼ • TFC - Más que postres | 19/09/13        | 8/01/14      | 105      |
| • PAC1: Plan de trabajo   | 19/09/13        | 1/10/13      | 13       |
| • PAC2: Análisis y diseño | 2/10/13         | 29/10/13     | 27       |
| • Implementación beta     | 30/10/13        | 26/11/13     | 27       |
| • PAC3: Implementación    | 27/11/13        | 10/12/13     | 12       |
| • Memoria y presentación  | 11/12/13        | 8/01/14      | 26       |

Figura 1

### 3.2 Planificación

Estas fases se subdividen en diversas tareas para poder llevarlas a término. Se ha tenido en cuenta la necesidad de dedicar tiempo a la investigación y aprendizaje en el uso de las nuevas tecnologías. Al disponer de un periodo de tiempo muy ajustado para llevar a término el proyecto se ha decidido trabajar todos los días de la semana. A continuación se muestra una tabla con las tareas incluidas en las diferentes fases del proyecto:

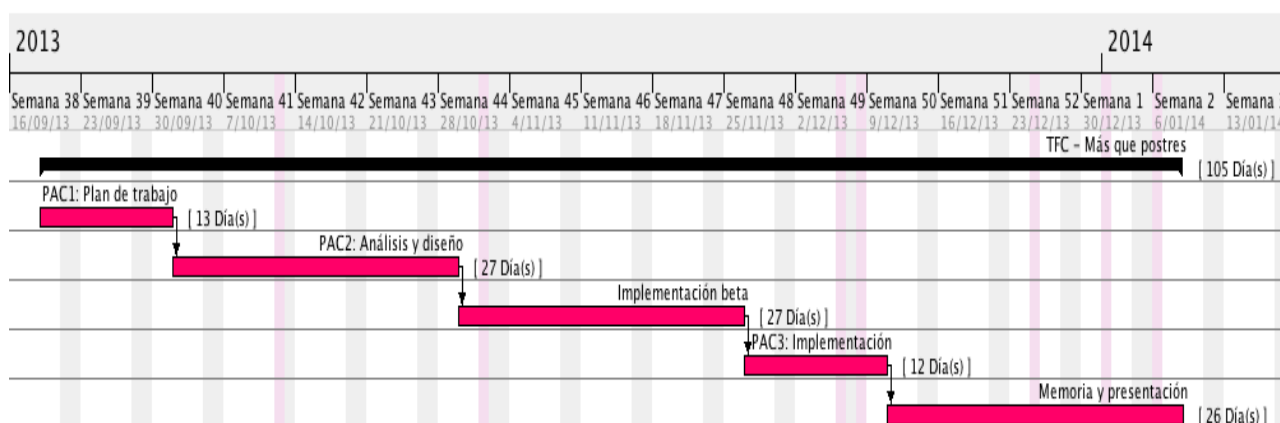


Figura 2

| TAREAS - PAC1: Plan de trabajo                        |
|---|
| Investigación inicial TFC: tecnologías, ideas, etc... |
| Propuesta proyecto al consultor                       |
| Encuentro virtual – 25/09/13                          |
| Preparación y redacción del Plan de trabajo           |
| Revisión y entrega del Plan de trabajo – 01/10/13     |

| <b>TAREAS - PAC2: Análisis y diseño</b>   |
|---|
| Estudio nuevas tecnologías  |
| Preparación entorno de desarrollo   |
| Pruebas iniciales con las nuevas tecnologías elegidas                           |
| Análisis funcionalidades de la aplicación                                       |
| Casos de uso: diagramas y descripción formal                                    |
| Diseño técnico: arquitectura de la aplicación, diseño de clases y base de datos |
| Diseño prototipos de IU   |
| Redacción del documento: Análisis y diseño                                      |
| Revisión y entrega del documento – 29/10/13                                     |

| <b>TAREAS - PAC3: Implementación (incluye fase implementación beta)</b> |
|---|
| Creación base de datos inicial  |
| Implementación módulo recetas   |
| Pruebas unitarias del módulo  |
| Implementación módulo favoritas   |
| Pruebas unitarias del módulo  |
| Implementación módulo compras   |
| Pruebas unitarias del módulo  |
| Implementación módulo información                                       |
| Pruebas unitarias del módulo  |
| Revisión de la implementación global                                    |
| Entrega implementación beta – 26/11/13                                  |
| Entrega implementación – 10/12/13                                       |

| <b>TAREAS – Entrega final: Memoria y video presentación</b>                      |
|--|
| Revisión y cumplimentación final del documento: Memoria                          |
| Aprendizaje realización video presentación                                       |
| Realización video presentación   |
| Entrega documentación final de proyecto: Memoria y video presentación – 08/01/14 |
| <b>Debate virtual posterior a la entrega final de todos los compañeros</b>       |

## 4 Riesgos del proyecto

| <b>Riesgo</b>   | <b>Probabilidad de suceso</b> | <b>Impacto</b> | <b>Acciones preventivas</b>                               |
|---|-------------------------------|----------------|---|
| Inexperiencia en desarrollo de aplicaciones para dispositivos móviles | Alta                          | Crítico        | Búsqueda de recursos formativos en internet, libros, etc. |



| Riesgo  | Probabilidad de suceso | Impacto | Acciones preventivas   |
|---|------------------------|---------|--|
| Falta de conocimientos y recursos para desarrollar el <i>backend</i>  | Media                  | Alto    | Investigación y posterior uso de un <i>BaaS</i> para minimizar la curva de aprendizaje y costes de un servidor.                    |
| Dispositivos móviles para realización de pruebas físicas inadecuados  | Media                  | Medio   | Préstamo por parte de familiares de móvil Android de mayor calidad, mini iPad con iOS7 y tableta Android.                          |
| Problemas técnicos con el ordenador y sus peligrosas consecuencias  | Baja                   | Alto    | Guardar en la nube todo el proyecto para evitar su pérdida y disponer de un ordenador portátil como alternativa en caso de avería. |
| Difícil conciliación del proyecto con la vida familiar como madre de 3 hijos, especialmente festivos y fines de semana. | Alta                   | Alto    | Dedicar al proyecto tiempo todos los días de la semana y siempre que sea posible avanzar para prevenir posibles contratiempos.     |

## 5 Análisis funcional

### 5.1 Funcionalidades de la aplicación

La aplicación tiene como finalidad básica facilitar al usuario el acceso a una recopilación de recetas que publica Angels Castelló en su blog [MAS QUE POSTRES](#). Por ello se ha decidido apostar por un diseño muy intuitivo y funcional de modo que no sea necesario dar instrucciones adicionales al usuario para comprender su manejo.

Para facilitar el acceso a las recetas, se han clasificado por categorías siguiendo la misma estructura que en su blog. Otra funcionalidad importante para el usuario es poder marcar las recetas que desee como favoritas para tener a posteriori un acceso más rápido a las mismas. Adicionalmente, otro funcionalidad de la aplicación es facilitar al usuario el acceso sin necesidad de tener conexión remota a la lista de la compra que contendrá los ingredientes que previamente haya seleccionado. Por último, se le facilitará al usuario el acceso al blog y *fan page* en *Facebook* de la autora.

### 5.2 Casos de usos

#### 5.2.1 Diagramas casos de usos

Se identifica un único actor principal en la aplicación, el cual es el usuario del dispositivo móvil, ya que no es una finalidad de la aplicación la gestión del *backend* de la misma. A continuación se puede ver un diagrama de los casos de uso identificados:

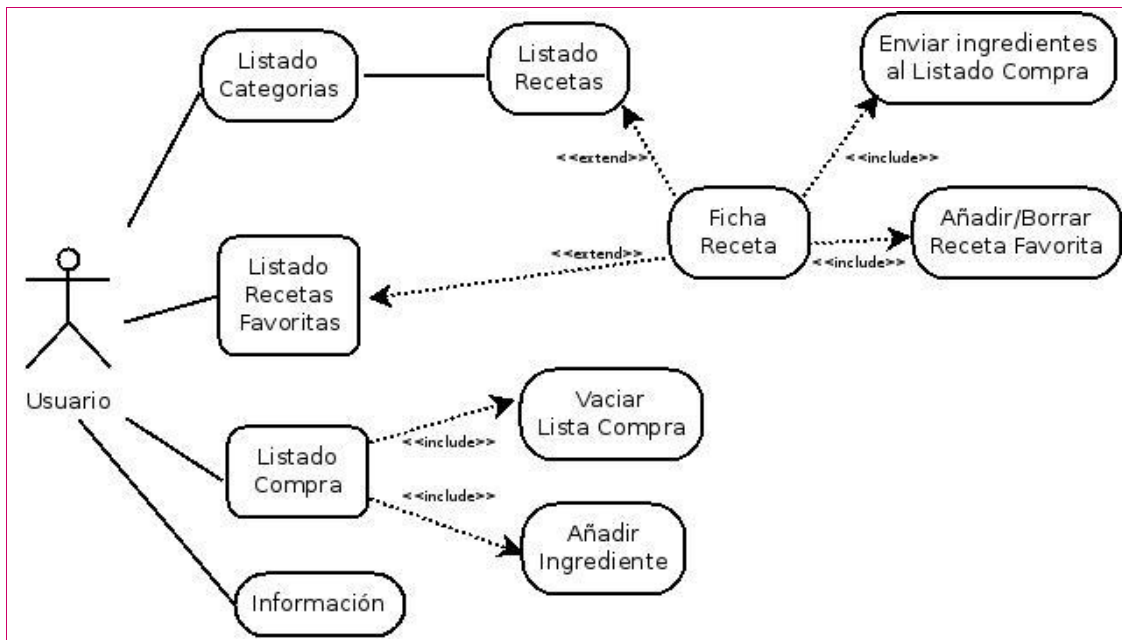


Figura 3

### 5.2.2 Descripción caso de uso [CU01] Listado Categorías

|                             |   |
|-----------------------------|---|
| <b>Identificador:</b>       | [CU01]  |
| <b>Nombre:</b>              | Listado categorías  |
| <b>Resumen:</b>             | Muestra listado de categorías en las que se encuentran categorizadas las recetas  |
| <b>Actores:</b>             | Usuario dispositivo móvil   |
| <b>Precondiciones:</b>      | Disponer de conexión remota   |
| <b>Postcondiciones:</b>     | Categoría seleccionada  |
| <b>Flujo normal:</b>        | El usuario selecciona una categoría del listado mostrado y el flujo de ejecución pasa a [CU02]  |
| <b>Flujos alternativos:</b> | 1) Si no se dispone de conexión a internet se mostrará un mensaje de aviso al usuario y finalizará el caso de uso<br>2) Finalizaría el caso de uso si durante el mismo el usuario decide seleccionar otra opción de navegación como pueden ser: [CU06], [CU07], [CU10]. |
| <b>Inclusiones:</b>         | Ninguna   |
| <b>Extensiones:</b>         | Ninguna   |

### 5.2.3 Descripción caso de uso [CU02] Listado Recetas

|                         |   |
|-------------------------|---|
| <b>Identificador:</b>   | [CU02]  |
| <b>Nombre:</b>          | Listado recetas   |
| <b>Resumen:</b>         | Muestra listado de recetas de la categoría seleccionada     |
| <b>Actores:</b>         | Usuario dispositivo móvil                                   |
| <b>Precondiciones:</b>  | 1) Disponer de conexión remota<br>2) Categoría seleccionada |
| <b>Postcondiciones:</b> | Receta seleccionada   |

|                             |   |
|-----------------------------|---|
| <b>Flujo normal:</b>        | El usuario tiene dos posibilidades:<br>1) selecciona una receta del listado mostrado y el flujo de ejecución pasa a [CU03]<br>2) selecciona el botón “atrás” y regresa a [CU01]   |
| <b>Flujos alternativos:</b> | 1) Si no se dispone de conexión a internet se mostrará un mensaje de aviso al usuario y finalizará el caso de uso<br>2) Finalizaría el caso de uso si durante el mismo el usuario decide seleccionar otra opción de navegación como pueden ser: [CU01], [CU06], [CU07], [CU10]. |
| <b>Inclusiones:</b>         | Ninguna   |
| <b>Extensiones:</b>         | [CU03]  |

### 5.2.4 Descripción caso de uso [CU03] Ficha Receta

|                             |   |
|-----------------------------|---|
| <b>Identificador:</b>       | [CU03]  |
| <b>Nombre:</b>              | Ficha receta  |
| <b>Resumen:</b>             | Muestra la ficha de la receta seleccionada  |
| <b>Actores:</b>             | Usuario dispositivo móvil   |
| <b>Precondiciones:</b>      | 1) Disponer de conexión remota<br>2) Receta seleccionada  |
| <b>Postcondiciones:</b>     | El usuario obtiene la información que ha seleccionado   |
| <b>Flujo normal:</b>        | El usuario tras consultar la ficha con sus ingredientes y preparación tiene tres opciones:<br>1) añadir/borrar la receta de favoritas (según flujo previo)<br>2) enviar los ingredientes de la receta a la lista de la compra<br>3) seleccionar el botón “atrás” y regresa a [CU02] |
| <b>Flujos alternativos:</b> | 1) Si no se dispone de conexión a internet se mostrará un mensaje de aviso al usuario y finalizará el caso de uso<br>2) Finalizaría el caso de uso si durante el mismo el usuario decide seleccionar otra opción de navegación como pueden ser: [CU01], [CU06], [CU07], [CU10].     |
| <b>Inclusiones:</b>         | [CU04] [CU05]   |
| <b>Extensiones:</b>         | Ninguna   |

### 5.2.5 Descripción caso de uso [CU04] Enviar ingredientes al L.Compra

|                             |   |
|-----------------------------|---|
| <b>Identificador:</b>       | [CU04]  |
| <b>Nombre:</b>              | Enviar ingredientes al listado de compra  |
| <b>Resumen:</b>             | Envía todos los ingredientes de la receta seleccionada a la lista de la compra  |
| <b>Actores:</b>             | Usuario dispositivo móvil   |
| <b>Precondiciones:</b>      | 1) Disponer de conexión remota<br>2) Receta seleccionada  |
| <b>Postcondiciones:</b>     | Ingredientes guardados en la lista de compra  |
| <b>Flujo normal:</b>        | El usuario tras seleccionar el botón “Ingredientes a la lista compra” ha generado el envío de todos los ingredientes de la receta a la lista de la compra. Se mantiene el flujo en [CU3]. Se pueden enviar tantas veces como se desee los ingredientes a la lista de la compra. |
| <b>Flujos alternativos:</b> | 1) Si no se dispone de conexión a internet se mostrará un mensaje   |

|                     |   |
|---------------------|---|
|                     | de aviso al usuario y finalizará el caso de uso |
| <b>Inclusiones:</b> | Ninguna   |
| <b>Extensiones:</b> | Ninguna   |

### 5.2.6 Descripción caso de uso [CU05] Añadir/Borrar Receta Favorita

|                             |   |
|-----------------------------|---|
| <b>Identificador:</b>       | [CU05]  |
| <b>Nombre:</b>              | Añadir/Borrar receta en listado favoritas   |
| <b>Resumen:</b>             | Añadir receta o Borrar receta del listado de favoritas  |
| <b>Actores:</b>             | Usuario dispositivo móvil   |
| <b>Precondiciones:</b>      | 1) Disponer de conexión remota<br>2) Receta seleccionada  |
| <b>Postcondiciones:</b>     | Receta añadida/borrada del listado de favoritas   |
| <b>Flujo normal:</b>        | Dependiendo de donde proceda el flujo previo a [CU03]:<br>1) Si procede de [CU02] se añadirá a favoritas. Sólo se puede añadir sino existía previamente en listado favoritas.<br>2) Si procede de [CU06] se borrará de favoritas.<br>Se mantiene el flujo en [CU3]. |
| <b>Flujos alternativos:</b> | 1) Si no se dispone de conexión a internet se mostrará un mensaje de aviso al usuario y finalizará el caso de uso   |
| <b>Inclusiones:</b>         | Ninguna   |
| <b>Extensiones:</b>         | Ninguna   |

### 5.2.7 Descripción caso de uso [CU06] Listado Recetas Favoritas

|                             |   |
|-----------------------------|---|
| <b>Identificador:</b>       | [CU06]  |
| <b>Nombre:</b>              | Listado recetas favoritas   |
| <b>Resumen:</b>             | Muestra listado de recetas favoritas  |
| <b>Actores:</b>             | Usuario dispositivo móvil   |
| <b>Precondiciones:</b>      | 1) Disponer de conexión remota<br>2) Opción menú Favoritas seleccionada   |
| <b>Postcondiciones:</b>     | Receta favorita seleccionada  |
| <b>Flujo normal:</b>        | El usuario selecciona una receta del listado mostrado y el flujo de ejecución pasa a [CU03]   |
| <b>Flujos alternativos:</b> | 1) Si no se dispone de conexión a internet se mostrará un mensaje de aviso al usuario y finalizará el caso de uso<br>2) Finalizaría el caso de uso si durante el mismo el usuario decide seleccionar otra opción de navegación como pueden ser: [CU01], [CU07], [CU10]. |
| <b>Inclusiones:</b>         | Ninguna   |
| <b>Extensiones:</b>         | [CU03]  |

### 5.2.8 Descripción caso de uso [CU07] Listado Compra

|                       |                |
|-----------------------|----------------|
| <b>Identificador:</b> | [CU07]         |
| <b>Nombre:</b>        | Listado Compra |

|                             |   |
|-----------------------------|---|
| <b>Resumen:</b>             | Muestra listado de los ingredientes a comprar y permite su gestión.   |
| <b>Actores:</b>             | Usuario dispositivo móvil   |
| <b>Precondiciones:</b>      | Opción menú Compra seleccionada   |
| <b>Postcondiciones:</b>     | El usuario obtiene/gestiona la información de la lista de la compra   |
| <b>Flujo normal:</b>        | El usuario tras consultar la lista de la compra tiene tres opciones:<br>1) borrar ingrediente de la lista pulsando icono “-” ubicado al lado del ingrediente. El flujo se mantiene en [CU07]<br>2) vaciar todos los ingredientes contenidos en lista de la compra pulsando el botón con esa leyenda. El flujo se mantiene en [CU07] tras ejecutar [CU08]<br>3) añadir ingrediente en la lista pulsando el botón con esa leyenda. El flujo pasa a [CU09] |
| <b>Flujos alternativos:</b> | Finalizaría el caso de uso si durante el mismo el usuario decide seleccionar otra opción de navegación como pueden ser: [CU01], [CU06], [CU10].   |
| <b>Inclusiones:</b>         | [CU08], [CU09]  |
| <b>Extensiones:</b>         | Ninguna   |

### 5.2.9 Descripción caso de uso [CU08] Vaciar Lista Compra

|                             |  |
|-----------------------------|--|
| <b>Identificador:</b>       | [CU08]   |
| <b>Nombre:</b>              | Vaciar listado compra  |
| <b>Resumen:</b>             | Borra todos los ingredientes contenidos en la lista de la compra.  |
| <b>Actores:</b>             | Usuario dispositivo móvil  |
| <b>Precondiciones:</b>      | 1) Opción vaciar lista de la compra seleccionada<br>2) La lista de la compra contiene ingredientes                           |
| <b>Postcondiciones:</b>     | Ingredientes eliminados (todos) de la lista de la compra   |
| <b>Flujo normal:</b>        | Se eliminan todos los ingredientes contenidos en la lista de la compra y el flujo retorna a [CU07] mostrando la lista vacía. |
| <b>Flujos alternativos:</b> | Ninguno  |
| <b>Inclusiones:</b>         | Ninguna  |
| <b>Extensiones:</b>         | Ninguna  |

### 5.2.10 Descripción caso de uso [CU09] Añadir Ingrediente

|                         |   |
|-------------------------|---|
| <b>Identificador:</b>   | [CU09]  |
| <b>Nombre:</b>          | Añadir ingrediente  |
| <b>Resumen:</b>         | Permite añadir ingrediente manualmente en la lista de la compra.  |
| <b>Actores:</b>         | Usuario dispositivo móvil   |
| <b>Precondiciones:</b>  | Opción añadir ingrediente seleccionada  |
| <b>Postcondiciones:</b> | Ingredientes añadido a la lista de la compra  |
| <b>Flujo normal:</b>    | Se abre cuadro de dialogo donde permite al usuario introducir un nuevo ingrediente manualmente a la lista de la compra. Tras introducirlo tiene dos opciones:<br>1) aceptar y se agrega el ingrediente a la lista.<br>2) cancelar con el botón “cancelar” o cerrando el cuadro de dialogo pulsando el icono “x” |

|                             |  |
|-----------------------------|--|
|                             | En ambos casos el flujo retorna a [CU07] mostrando la lista actualizada según sea el caso. |
| <b>Flujos alternativos:</b> | Ninguno  |
| <b>Inclusiones:</b>         | Ninguna  |
| <b>Extensiones:</b>         | Ninguna  |

### 5.2.11 Descripción caso de uso [CU10] Información

|                             |  |
|-----------------------------|--|
| <b>Identificador:</b>       | [CU10]   |
| <b>Nombre:</b>              | Información  |
| <b>Resumen:</b>             | Muestra información adicional sobre la autora de las recetas   |
| <b>Actores:</b>             | Usuario dispositivo móvil  |
| <b>Precondiciones:</b>      | 1) Disponer de conexión remota<br>2) Opción menú Info seleccionada   |
| <b>Postcondiciones:</b>     | El usuario obtiene información y acceso a enlaces externos   |
| <b>Flujo normal:</b>        | El usuario obtiene información sobre la autora y tiene acceso directo a dos enlaces externos:<br>1) Blog de Mas Que Postres.<br>2) Fan Page en Facebook.<br>En ambos casos, el flujo interno de la aplicación se rompe abriendo una nueva ventana del navegador que accede a las direcciones URL de los enlaces. |
| <b>Flujos alternativos:</b> | Finalizaría el caso de uso si durante el mismo el usuario decide seleccionar otra opción de navegación como pueden ser: [CU01], [CU06], [CU07].  |
| <b>Inclusiones:</b>         | Ninguna  |
| <b>Extensiones:</b>         | Ninguna  |

## 6 Diseño técnico

### 6.1 Arquitectura de la aplicación

La arquitectura física de la aplicación se ha basado en una arquitectura cliente-servidor, donde el cliente es todo dispositivo móvil en el que se ejecute la aplicación y para el servidor se ha contado con *Parse* como *BaaS* en la nube (ver información en punto 6.1.3.1.). Es muy importante disponer de una conectividad continua (independientemente de que sea *wifi* o *3G*, *4G*) ya que la aplicación ha sido diseñada así. No se ha pretendido en ningún momento que pueda trabajar *offline*. Sólo se ha diseñado una pequeña parte de la misma para trabajar sin conexión a internet con fines docentes.

Se ha tratado de aplicar el patrón **MVC** (*model*, *view*, *controller*) en la arquitectura lógica de la aplicación. A continuación se detalla cada capa y su implementación en la aplicación:

#### 6.1.1 Capa de presentación (vista)

Es la interfaz gráfica de nuestra aplicación, la que nos permite interactuar con el usuario y



presentarle los datos. Para ello utilizamos *HTML5*, *CSS3* y nos ayudamos principalmente del framework *jQuery Mobile*.

### 6.1.2 Capa de negocio (controlador)

Es donde se determina que vistas se muestran al usuario. Se encarga de interactuar entre la capa de datos y la capa de presentación. En nuestra aplicación controlamos la lógica de negocio en los ficheros JavaScript.

### 6.1.3 Capa de datos (modelo)

Es la capa que accede a los datos solicitados por la capa de negocio. En este caso, haremos uso de la API *LocalStorage* de *HTML5* para el almacenamiento en local y por otro lado nos ayudamos de un servicio de *backend* como es *Parse* para el que necesitaremos tener conexión remota.

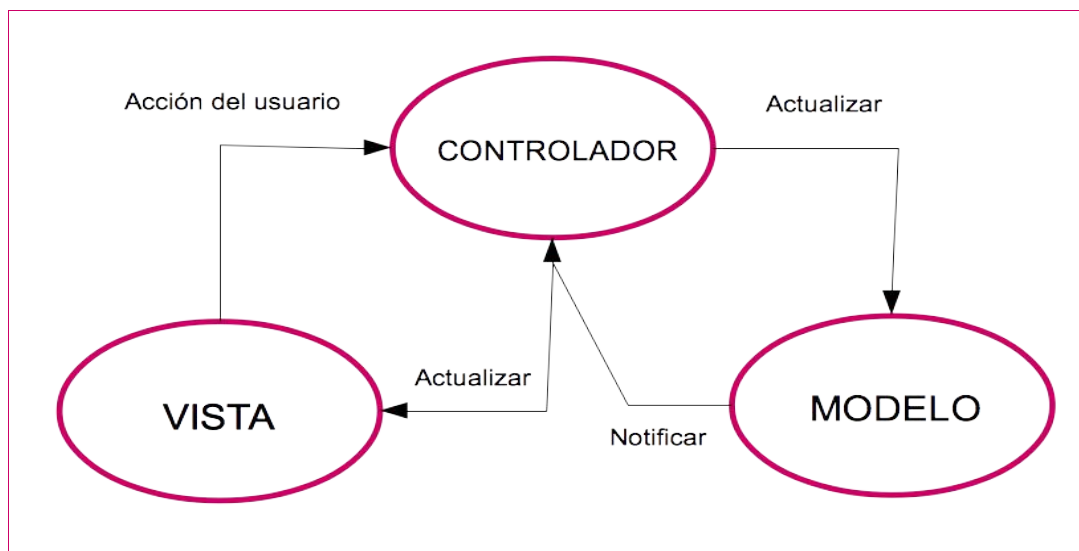


Figura 4

#### 6.1.3.1 Parse como BaaS

Finalmente se utilizará *Parse* como *BaaS* dado que se ajusta más al perfil de las necesidades del proyecto. En la decisión también ha pesado que *Parse* cuenta con el sólido respaldo de *AWS (Amazon Web Services)* como proveedor en la nube. *Parse* se basa en seis pilares:

- Parse Data: Permite almacenar los datos de la aplicación en la nube y de este modo ya no es necesario disponer de un servidor.
- Parse Push: Permite crear, planificar y segmentar las notificaciones *push* muy fácilmente.
- Parse Social: Permite integrar nuestra aplicación con las redes sociales y registrar a los usuarios con sus cuentas de *facebook* y *twitter* sin tener que programar grandes cantidades de código.

- **Parse Hosting:** Permite crear web apps multi-plataforma con acceso a *Parse Data* utilizando su *JavaScript SDK*. También, usando el *framework Express*, permite crear web apps dinámicas que utilizan el servicio *Cloud Code*.
- **Parse Analytics:** Permite monitorizar y analizar nuestra aplicación.
- **Cloud Code:** Tanto en el lado cliente como en la nube se utiliza *Parse JavaScript SDK* para acceder a los datos. Ya no es necesario servidores para el *backend* utilizando la nube. Permite gestionar las versiones del código que se desarrolla y despliega. El código corre en *Parse Cloud* en vez de en el dispositivo móvil, de este modo si se actualiza en *Cloud Code* instantáneamente está disponible en todos los dispositivos móviles.

El servicio *Basic* de *Parse* es ideal para nuestro proyecto ya que ofrece las siguientes características:

- Gratuito
- 1 millón de peticiones (*API requests*) al mes
- 1GB Almacenaje de ficheros
- 1 millón de notificaciones (*pushes*) al mes
- *Parse Core Platform*

## 6.2 **Diseño base de datos**

Con el fin de lograr el objetivo docente de conocer y aplicar diferentes técnicas de persistencia de datos, en la aplicación haremos uso de dos tipos distintos de almacenaje: el almacenaje que realizaremos en la nube a través de **Parse** para el cual necesitamos tener conexión de internet y el almacenaje local de datos que realizaremos en el propio dispositivo móvil gracias al API de *JavaScript* que incluye *HTML5* y que permite interactuar con **localStorage**.

### 6.2.1 **Parse Data**

Al utilizar el servicio **Parse Data**, el almacenaje de datos se construye sobre un objeto (*Parse.Object*) que contiene clave y valor (*key-value pairs*) cuyo formato es totalmente compatible con los datos en formato JSON. El servicio *Parse Data* implementa la interface *Backbone.Model* para los objetos.

También permite crear colecciones de objetos mediante *Parse.Collection*, las cuales son compatibles con *Backbone.Collection* y además implementa las mismas funcionalidades.

Para almacenar imágenes se utilizará *Parse.File* que permite almacenar ficheros en la nube de hasta 10Mb de tamaño.

*Parse Data* cuenta con un excelente panel (*dashboard*) que facilita la introducción de los datos en el *backend* de la aplicación.

**CLASE: CategoriaObject**

| Key                                    | Value   |
|--|---|
| <b>ObjectId</b><br><i>tipo: String</i> | "TzqtqpEau3" (Autogenerado por <i>Parse</i> ) |
| <b>Titulo</b><br><i>tipo: String</i>   | "Tartas"                                      |
| <b>Foto</b><br><i>tipo: File</i>       | tarta merengue.JPG                            |

**CLASE: RecetaObject**

| Key                                       | Value  |
|---|--|
| <b>ObjectId</b><br><i>tipo: String</i>    | "OK1iwHv2v" (Autogenerado por <i>Parse</i> )   |
| <b>Titulo</b><br><i>tipo: String</i>      | "Magdalenas rellenas de compota de manzana"  |
| <b>Preparacion</b><br><i>tipo: Array</i>  | ["1-Montamos los huevos con el azúcar, y el aroma de vainilla. Vamos añadiendo aceite poco a poco pero de forma seguida.",<br>"2-Mezclamos la levadura con la harina y lo tamizamos. Lo agregamos a la mezcla anterior, removiéndolo bien.",<br>"3-Llenamos 3/4 de las cápsulas para magdalenas. Rellenamos una manga pastelera con la compota de manzana y introducimos la punta de la manga en la masa e inyectamos un poco de compota a gusto.....",<br>"4- Horneamos a 200°C durante 10 min."] |
| <b>Ingredientes</b><br><i>tipo: Array</i> | ["250g de huevos enteros", "250 g de azúcar", "250 g de aceite de girasol", "250 g de harina floja", "10g de levadura química", "aroma de vainilla", "compota de manzana"]   |
| <b>Categorias</b><br><i>tipo: Array</i>   | ["Desayunos", "Muffins"]   |
| <b>Foto</b><br><i>tipo: File</i>          | manzanas rellenas de compota.JPG   |

**6.2.2 LocalStorage**

PhoneGap da soporte a la opción de almacenaje local (*LocalStorage*). La persistencia de los datos almacenados con *localStorage* es definitiva hasta que se elimine expresamente. Los datos almacenados utilizando *localStorage* se guardan en parejas de clave/valor (*key-value pairs*) y sólo se permite guardar cadenas de texto. Esto nos permite encapsular los datos como objetos JSON.

A continuación vemos un ejemplo de código de como guardar un valor:

```
localStorage.setItem("key_name", value);
```

Se ha decidido utilizar la opción de almacenaje local para guardar y gestionar los ingredientes en la lista de la compra, tanto los ingredientes contenidos en las recetas como los creados manualmente por el usuario. También para guardar las recetas favoritas del usuario.

**TABLA: Compra**

| Key                                       | Value            |
|---|------------------|
| <b>Ingrediente</b><br><i>tipo: String</i> | "250g de azúcar" |

**TABLA: Favoritas**

| Key                              | Value   |
|----------------------------------|---|
| <b>id</b><br><i>tipo: String</i> | "OK1iwHv2vv" (Corresponde al <b>ObjectId</b> de <b>RecetaObject</b> en <b>Parse</b> ) |

**6.3 Diagramas de clases**

A continuación se adjunta el diagrama de clases global de la aplicación, independientemente del tipo de almacenaje. Cabe recalcar que en el diseño de los datos a almacenar, cuando nos referimos a Ingrediente hay dos posibles tipos: los contenidos en una receta y los ingredientes creados manualmente. Dicha distinción no se ha reflejado en el diagrama de clases.

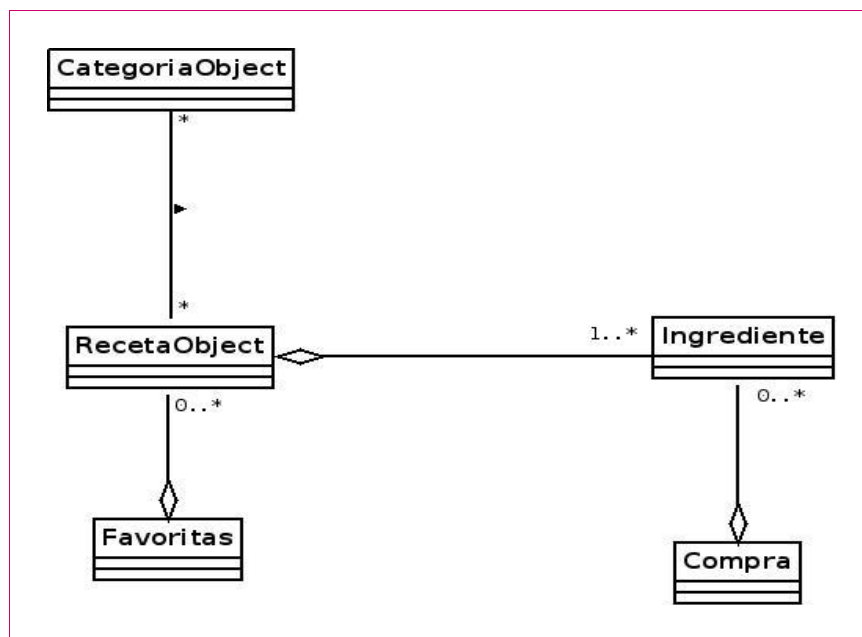


Figura 5

**7 Prototipo**

Para realizar los prototipos de las pantallas, tras valorar las diferentes opciones existentes en el mercado, se ha decidido utilizar **Codiqa** ya que es una excelente herramienta *online* que dispone de una amplia librería de componentes *jQuery Mobile*. Además permite importar, editar y exportar el código *HMTL+CSS+JS* muy cómodamente.

*Codiqa* dispone de licencias académicas para estudiantes, las cuales son gratuitas y deben ser renovadas cada seis meses.

## 7.1 Pantallas: Home y RecetasPage

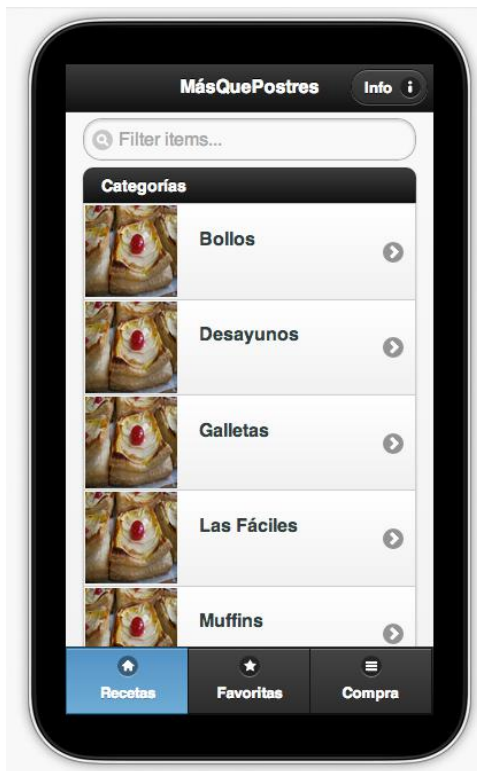


Figura 6 – Home



Figura 7 – RecetasPage

## 7.2 Pantalla: RecetasFicha



Figura 8 – RecetasFicha

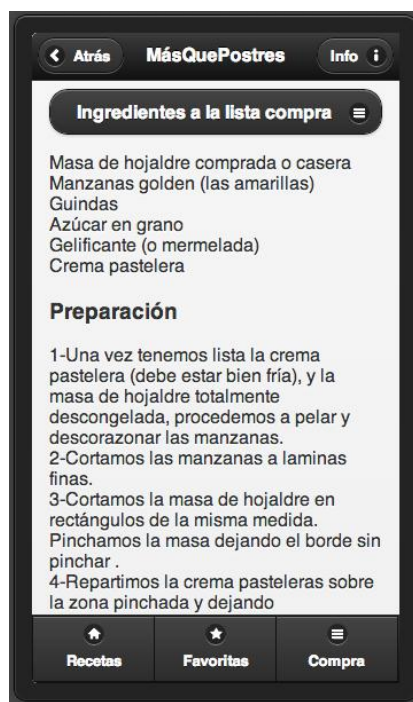


Figura 9 – RecetasFicha

### 7.3 ***Pantalla: FavoritasPage***



Figura 10 – FavoritasPage

### 7.4 ***Pantalla: FavoritasFicha***



Figura 11 – FavoritasFicha

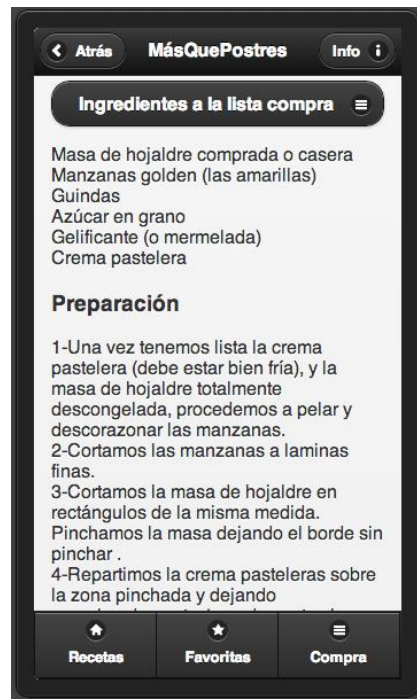


Figura 12 – FavoritasFicha



### 7.5 **Pantalla: ComprasPage y DialogoIngrediente**



Figura 13 – ComprasPage



Figura 14 – DialogoIngrediente

### 7.6 **Pantalla: InfoPage**



Figura 15 – InfoPage

### 7.7 Flujo de navegación

Tal como se indica en la leyenda de la siguiente figura, todas las pantallas que contienen la barra de navegación inferior tienen también acceso directo a las pantallas Home, FavoritasPage y ComprasPage. Adicionalmente, todas ellas a excepción de InfoPage, al disponer del botón icono información, también tienen acceso directo a InfoPage.

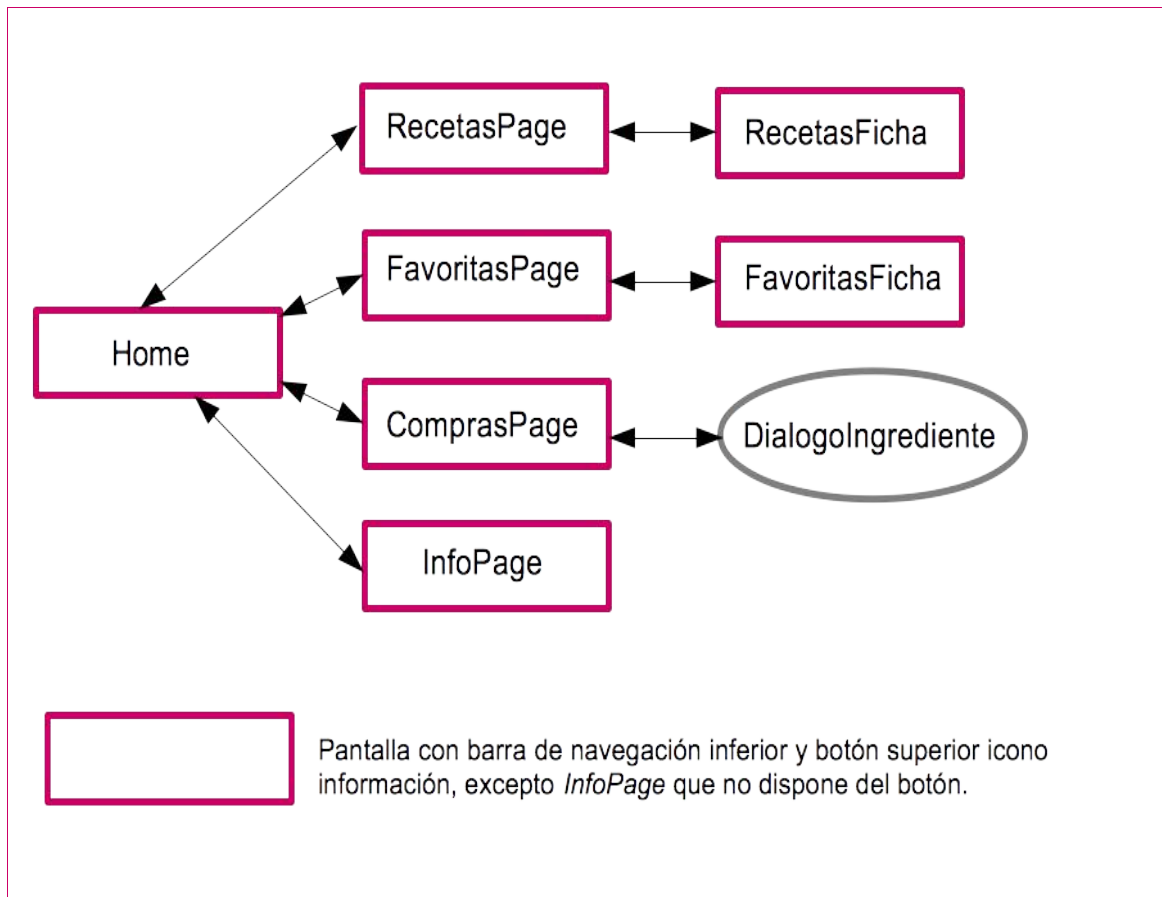


Figura 16

## 8 Implementación y pruebas

### 8.1 Código HTML y JS

Para la implementación se ha empleado el editor de código **Brackets** que permite depurar todo el código en vivo en el navegador Google Chrome. Además se ha utilizado la extensión **Theseus for Brackets** de Tom Lieber que permite depurar retroactivamente JavaScript en Chrome dentro del propio editor.

A continuación se muestra como se han llamado a los estilos, librerías y ficheros JS necesarios para la implementación de la aplicación. En los comentarios se puede ver la explicación de su uso.

```

1 <!doctype html>
2 <html lang="es">
3
4 <head>
5   <title>Más Que Postres</title>
6   <meta name="viewport" content="user-scalable=no, initial-scale=1, width=device-width">
7   <meta http-equiv="content-type" content="text/html; charset=UTF-8">
8   <!-- Estilos -->
9   <link rel="stylesheet" href="css/jquery.mobile-1.2.1.min.css" />
10  <link rel="stylesheet" href="css/estilos.css" />
11  <!-- Librerías JS -->
12  <script src="js/jquery-1.8.3.min.js"></script>
13  <script src="js/jquery.mobile-1.2.1.min.js"></script>
14  <!-- Para que Phonegap Build pueda construir la app -->
15  <script type="text/javascript" src="cordova.js"></script>
16  <!-- Baas Parse -->
17  <script type="text/javascript" src="http://www.parsecdn.com/js/parse-1.2.12.min.js"></script>
18  <!-- JS de la aplicación -->
19  <script type="text/javascript" src="js/main.js"></script>

```

El proyecto se ha basado en la utilización de jQuery Mobile para su implementación de manera que se ha seguido una estructura con múltiples páginas dentro de un único fichero HTML llamado **index.html**. Cada bloque página sigue esta estructura y tiene un identificador **id** único.

```

<div data-role="page" id="...">
  <div data-role="header">...</div>
  <div data-role="content">...</div>
  <div data-role="footer">...</div>
</div>

```

Como se puede ver en el extracto de muestra de una de esas páginas, parte del contenido será generado de forma dinámica. Lo mismo ocurre en prácticamente el resto de páginas.

```

<body onload="onLoad()">
  <!-- home -->
  <div data-role="page" id="home">
    <div data-theme="a" data-role="header" data-position="fixed">
      <a data-role="button" href="#infoPage" data-icon="info" data-iconpos="right" class="ui-btn-right">Info</a>
      <h3>MásQuePostres</h3>
    </div>
    <!-- /header -->
    <div data-role="content">
      <!-- Se generara el contenido de forma dinámica-->
    </div>
    <!-- /content -->
    <div data-role="footer" data-position="fixed">
      <div data-role="navbar" data-iconpos="top" data-theme="a">
        <ul>
          <li>
            <!-- Boton activo: class="ui-btn-active" -->
            <a href="#" data-transition="slide" data-icon="home" class="ui-btn-active">Recetas</a>
          </li>
          <li>
            <a href="#favoritasPage" data-transition="slide" data-icon="star">Favoritas</a>
          </li>
          <li>
            <a href="#comprasPage" data-transition="slide" data-icon="bars">Compra</a>
          </li>
        </ul>
      </div>
    </div>
    <!-- /footer -->
  </div>
  <!-- /page home -->

```

Para generar y inyectar contenido a estas páginas dinámicamente llamaremos manualmente a `$.mobile.changePage()`.

```
// Listen cualquier intento de llamar a changePage().
$(document).bind("pagebeforechange", function (e, data) {
  if (typeof data.toPage === "string") {
    // Solo manejamos un subset de URLs.
    var u = $.mobile.path.parseUrl(data.toPage);
    var goHome = /^#home/;
    var goRecetasPage = /^#recetasPage/;
    var goRecetasDetail = /^#recetasFicha/;
    var goFavoritasPage = /^#favoritasPage/;
    var goFavoritasDetail = /^#favoritasFicha/;
    var goComprasPage = /^#comprasPage/;

    if (u.hash.search(goHome) !== -1) {
      // Mostrar listado categorias
      showCategoriasList(u, data.options);
      e.preventDefault();
    } else if (u.hash.search(goRecetasPage) !== -1) {
      // Mostrar listado recetas para la categoria seleccionada
      showRecetasList(u, data.options);
      e.preventDefault();
    } else if (u.hash.search(goRecetasDetail) !== -1) {
      // Mostrar ficha de la receta seleccionada
      showRecetasDetail(u, data.options);
      e.preventDefault();
    } else if (u.hash.search(goFavoritasPage) !== -1) {
      // Mostrar listado recetas favoritas
      showFavoritasList(u, data.options);
      e.preventDefault();
    } else if (u.hash.search(goFavoritasDetail) !== -1) {
      // Mostrar ficha de la receta FAVORITA seleccionada
      // Reutilizar showRecetasDetail
      showRecetasDetail(u, data.options);
      e.preventDefault();
    } else if (u.hash.search(goComprasPage) !== -1) {
      // Mostrar listado compra y su gestión
      showComprasList(u, data.options);
      e.preventDefault();
    }
  }
});
```

En la siguiente página del documento se muestra la función `showCategoriasList()` como ejemplo de la generación e inyección del contenido en la página dinámicamente. En los comentarios se ha detallado el proceso para su mejor comprensión.

En esta función vemos como se realiza una consulta al *BaaS Parse* para obtener los objetos *CategoriaObject* ordenado por el *título* de forma ascendente.

Durante la fase de desarrollo de la aplicación se ha utilizado `console.log()` para hacer un seguimiento del correcto funcionamiento del código.

Todo el código JavaScript creado durante la implementación de la aplicación se ha guardado en un único fichero llamado **main.js**.

```

function showCategoriasList(urlObj, options) {
  // Obtenemos la pag en la que vamos a escribir nuestro contenido
  var $page = $("#home");
  var $content = $page.children(":jqmData(role=content)");
  // Construimos la lista dinámica
  var markup = '<ul id="categoriasList" data-theme="c" data-role="listview" data-inset="true" data-filter="true"
data-filter-placeholder="¿Qué categoría buscas?">';
  markup += '<li data-theme="a" data-role="list-divider" role="heading">Categorías</li>';
  // Petición a Parse
  var query = new Parse.Query(CategoriaObject);
  // En orden ascendente de Título
  query.ascending("Titulo");
  query.find({
    success: function (results) {
      console.log(results);
      for (var i = 0, len = results.length; i < len; i++) {
        var categoria = results[i];
        console.log(categoria);
        var pic = categoria.get("Foto");
        markup += '<li><a href="#recetasPage?category=' + categoria.get('Titulo') + '" data-
transition="slide"><h3>' + categoria.get('Titulo')
+ '</h3></a></li>';
      }
      markup += '</ul>';
      console.log('getCategorias en el Query' + markup);
      // Inyectamos la lista construida
      $content.html(markup);
      // Las pags son cargadas perezosamente. Llamamos a page() en el elemento page para estar seguros de que
es cargada antes de recargar el listview markup que hemos inyectado
      $page.page();
      // Cargamos el listview que hemos inyectado
      $content.find(":jqmData(role=listview)").listview();
      // Nos aseguramos que incluye/muestra los parametros la url
      options.dataUrl = urlObj.href;
      console.log('options.dataUrl =' + options.dataUrl);
      // Llamamos a changePage() y le decimos que cambie a la pag que justo hemos modificado
      $.mobile.changePage($page, options);
    },
    error: function (error) {
      alert("Error when getting categories!");
    }
  });
}
}

```

## 8.2 ***PhoneGap Build***

Una vez implementadas las funcionalidades básicas de la aplicación, se ha procedido a construir nuestra aplicación con **PhoneGap Build**. Nos hemos registrado a través de una cuenta de **Github** y siguiendo las instrucciones detalladas en la documentación de **PhoneGap Build** hemos creado el fichero **config.xml** y todos los restantes ficheros necesarios (iconos y *splash* para **Android** y **iOS**). Esto último ha hecho que aumentase considerablemente el peso de nuestra aplicación.

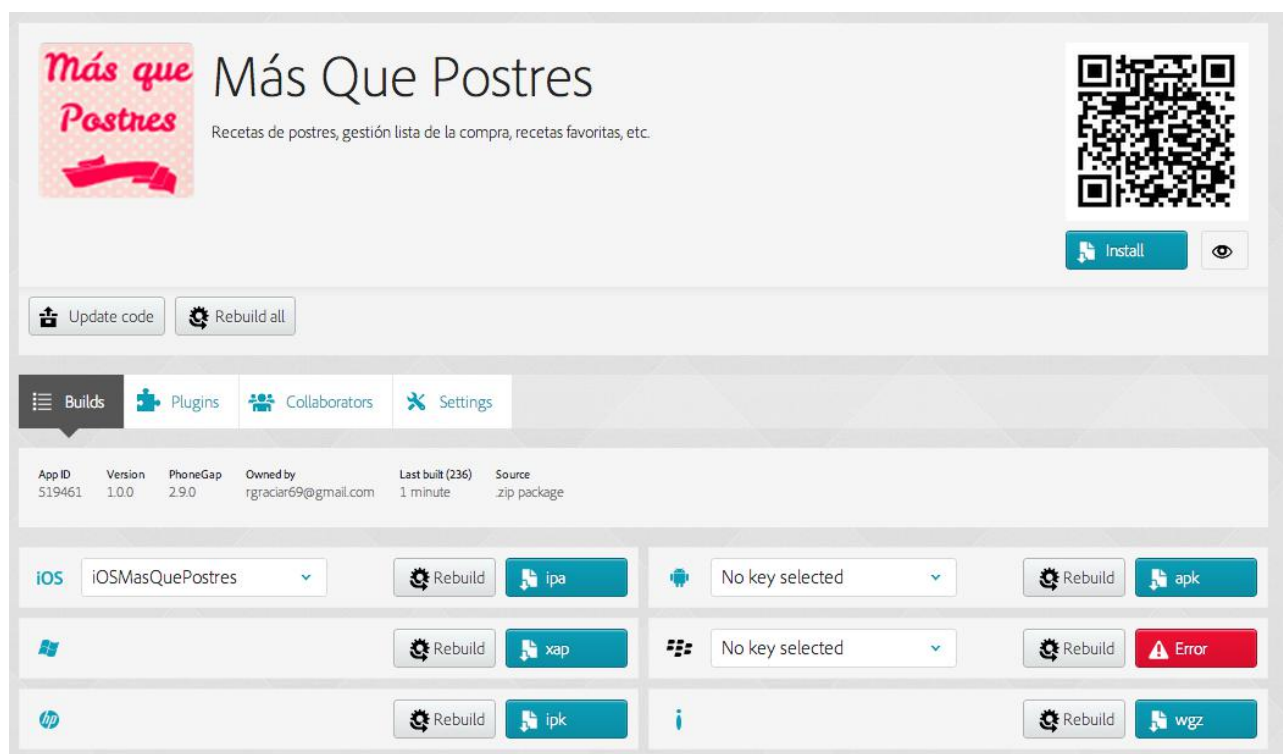
**PhoneGap Build** permite construir nuestra aplicación para múltiples plataformas, pero en el caso de **iOS**, es necesario registrarse con un **certificado de desarrollador** y un **provisioning profile**. Obtener ambos es una tarea algo laboriosa a través de la web de **Apple** y su **iOS Developer Program**. Finalmente se ha obtenido, pero sólo para desarrollo y no para producción.

La construcción de la aplicación tras estos pasos antes mencionados, resulta muy sencilla. Se sube el fichero comprimido de la aplicación a través del botón *Update code* y automáticamente



empiezan a generarse la aplicación para las distintas plataformas. Una vez concluido el proceso, se puede descargar el fichero en concreto para la plataforma deseada o aún más cómodo, escanear con nuestro dispositivo móvil el código QR generado.

En esta imagen podemos ver como queda tras la construcción por parte de **PhoneGap Build**. Si nos fijamos, se puede observar que se han realizado 236 construcciones donde pone *Last built*, esto es porque podemos subir y generar tantas veces como queramos la aplicación pero hay que tener en cuenta que es una acción destructiva del anterior código subido.



Se ha decidido que tras comprobar que el dispositivo está listo, en nuestro código JS se verifique que se dispone de conexión a internet. En el caso de que no se disponga de conexión se mostrará un mensaje al usuario de la aplicación informándole. En este mensaje también se le comunica que sólo podrá trabajar con la opción Compra si decide continuar. Por ello, se han desactivado el resto de opciones de navegación cuando ocurre este caso.

En el caso de que se disponga conexión, se quedará escuchando por si en cualquier momento se perdiera la misma y pasaría a tener sólo la opción Compra en las condiciones antes mencionadas.

Para comprobar su correcto funcionamiento se han realizado una serie de pruebas con la aplicación instalada en tres dispositivos móviles diferentes:

**iPad Mini 16GB con iOS 7.0.4**

**Tableta Acer modelo A1-810 con Android 4.2.2**

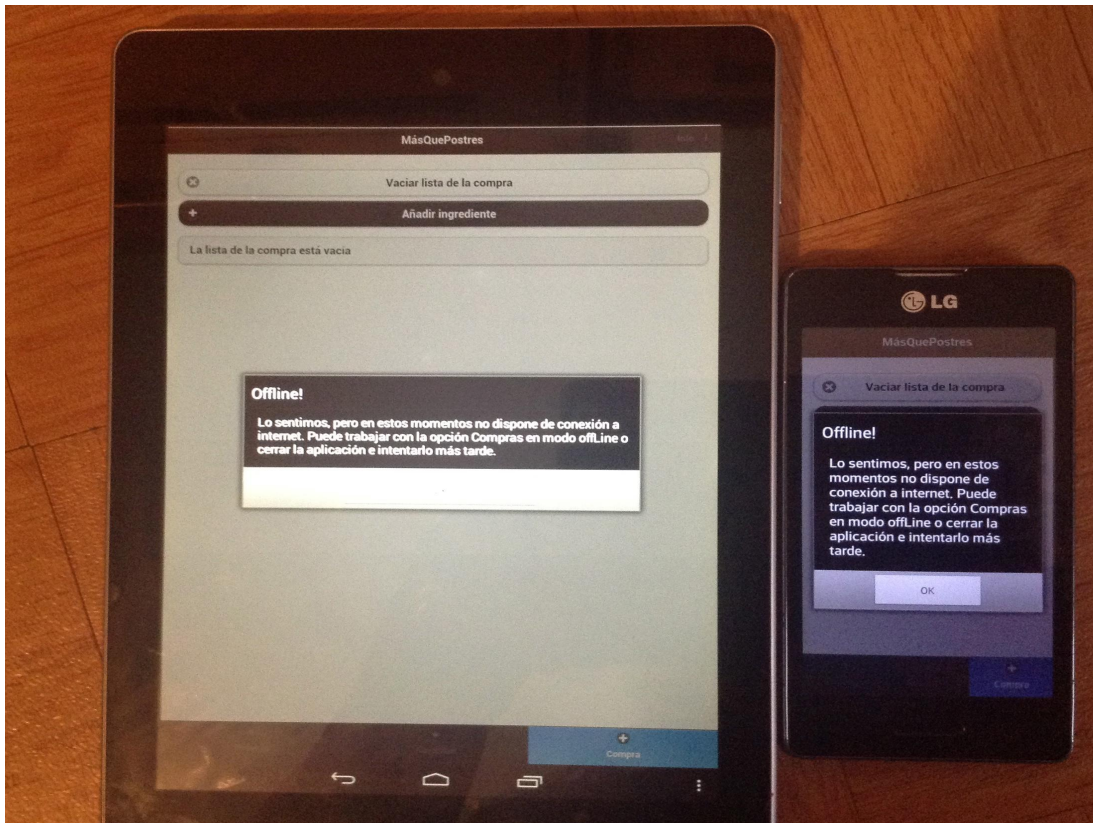
**Móvil LG Optimus L5 II con Android 4.1.2**



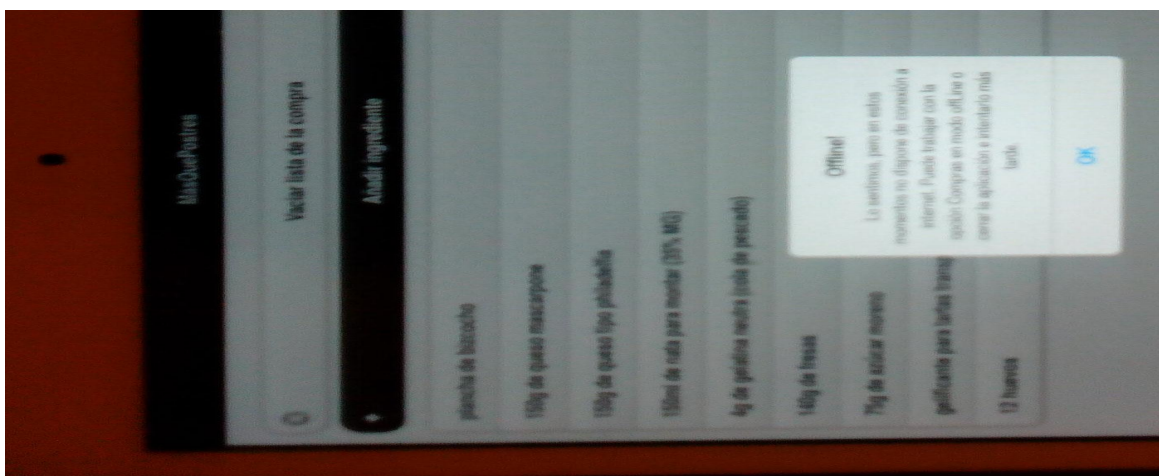
A continuación se muestran varias capturas de pantalla de la aplicación en los tres dispositivos.

### 8.3 Capturas aplicación

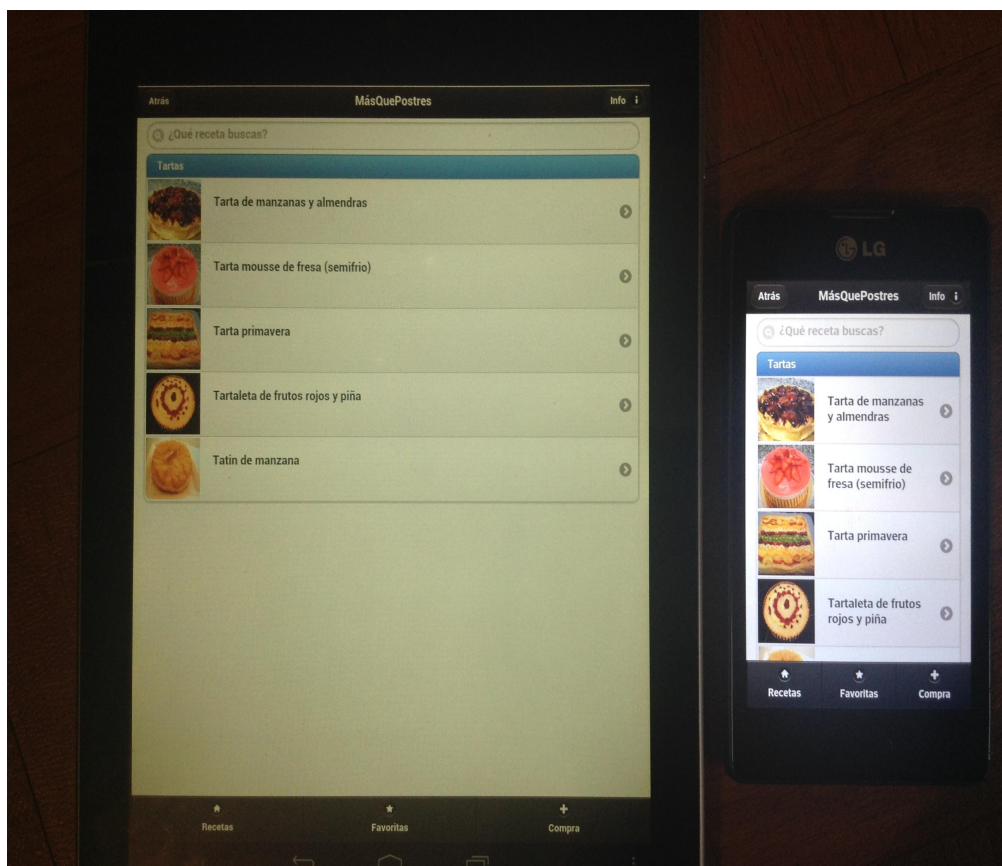
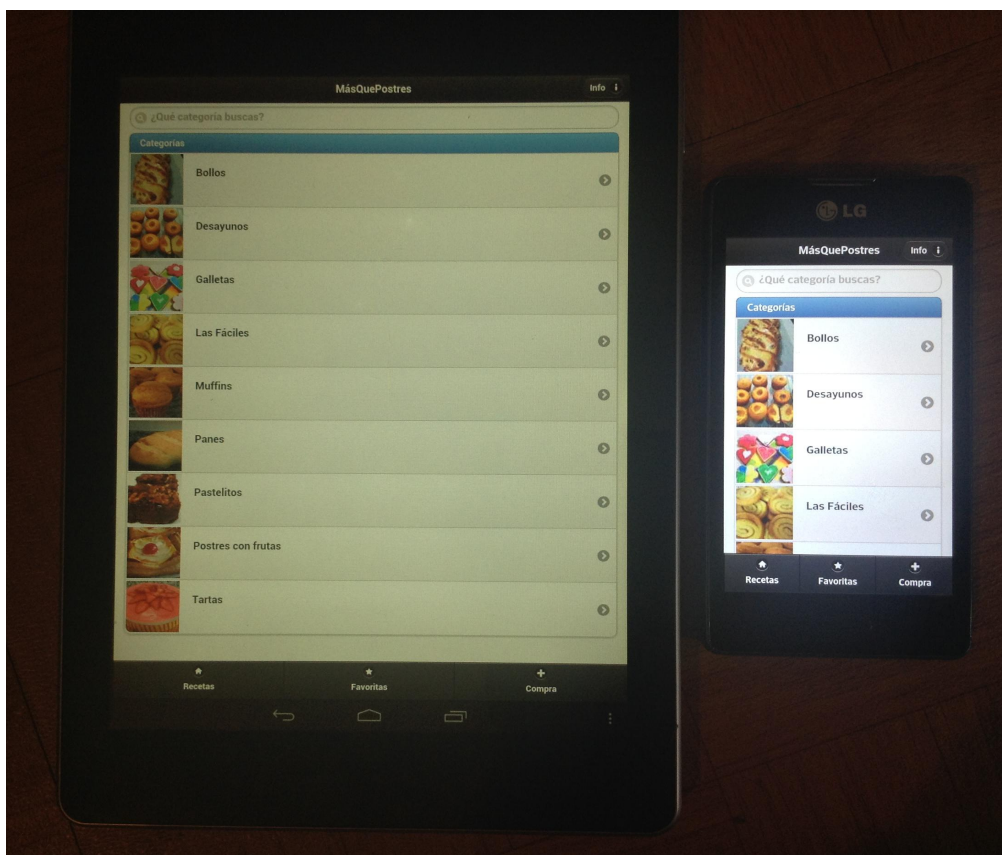
Esta primera captura muestra el caso en que se detecta que no hay conexión de internet y se avisa al usuario, además de mostrar sólo la opción de Compra con las otras opciones de navegación desactivadas.



En iOS, dada la mala calidad de la fotografía sólo se muestra la parte interesante con el mensaje al usuario.

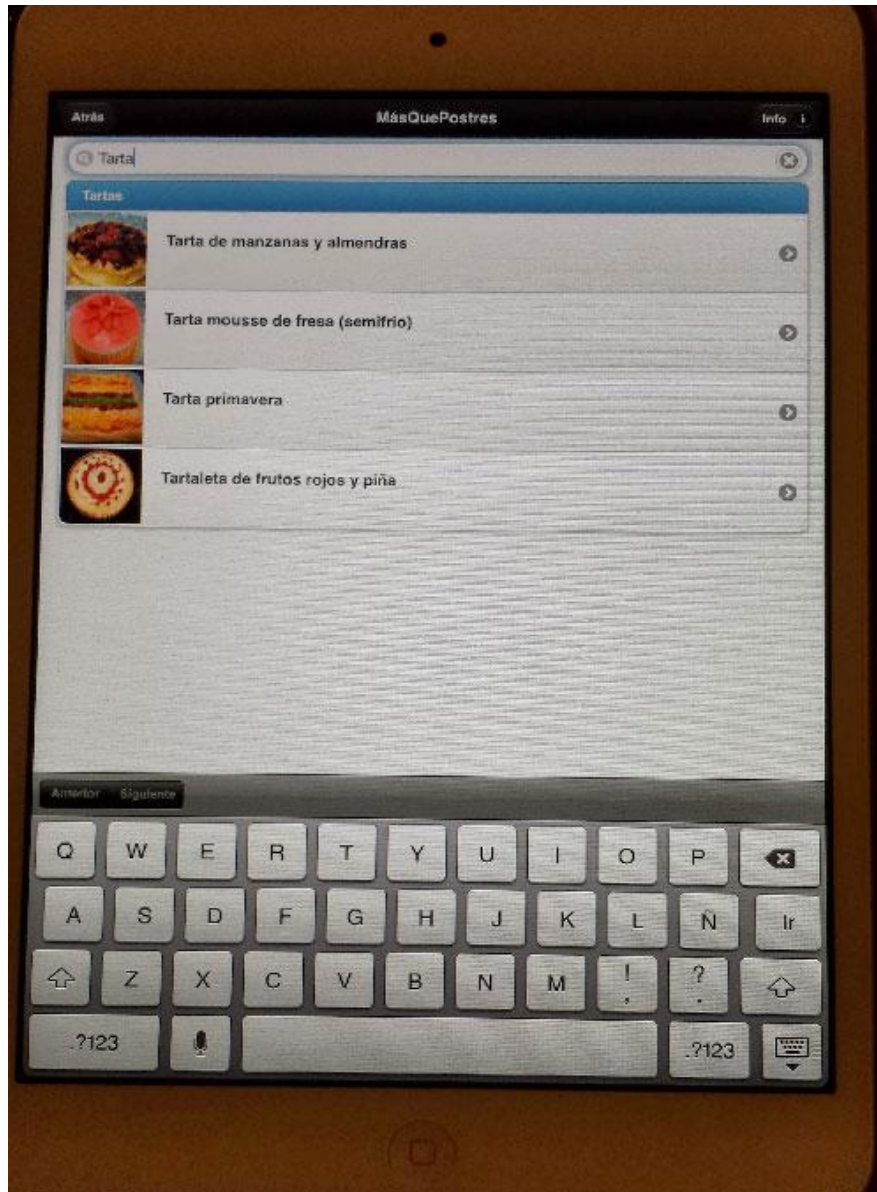


La pantalla inicial que muestra las categorías es la siguiente y a continuación la pantalla de las recetas correspondientes a la categoría seleccionada. En este caso **Tartas**.



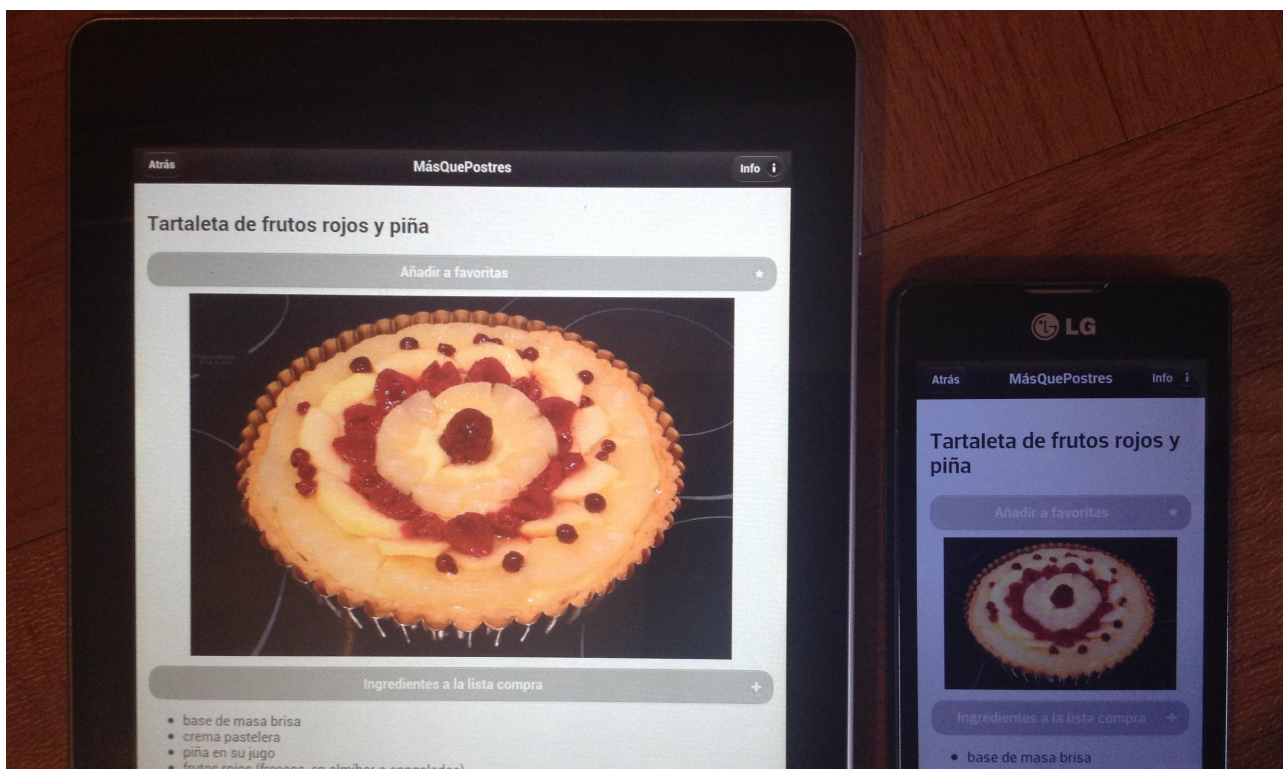
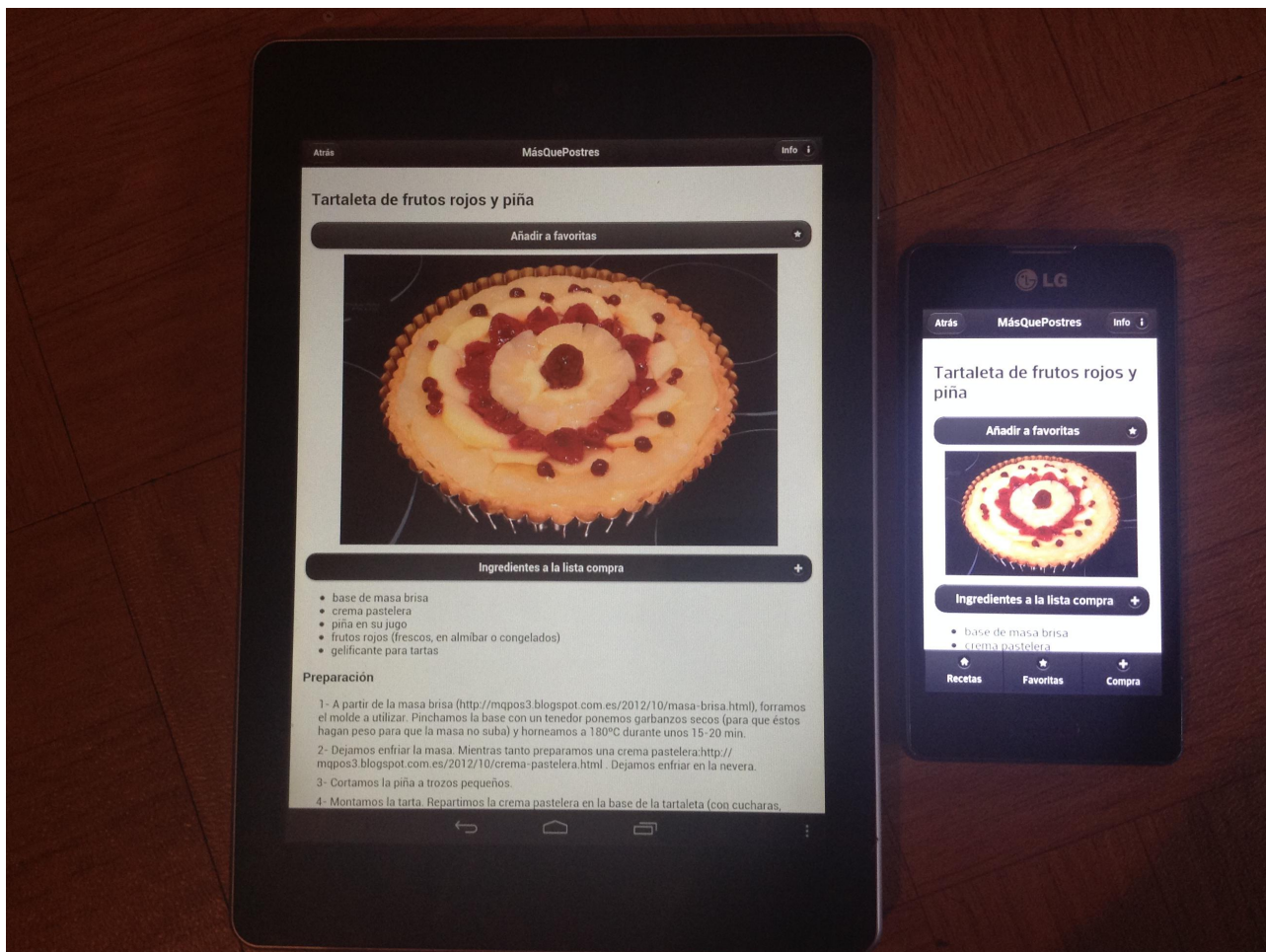


Ambas pantallas anteriores tienen la opción de buscar en la parte superior. En esta imagen podemos ver la búsqueda de las recetas que empiezan por **Tarta** en el mini iPad.



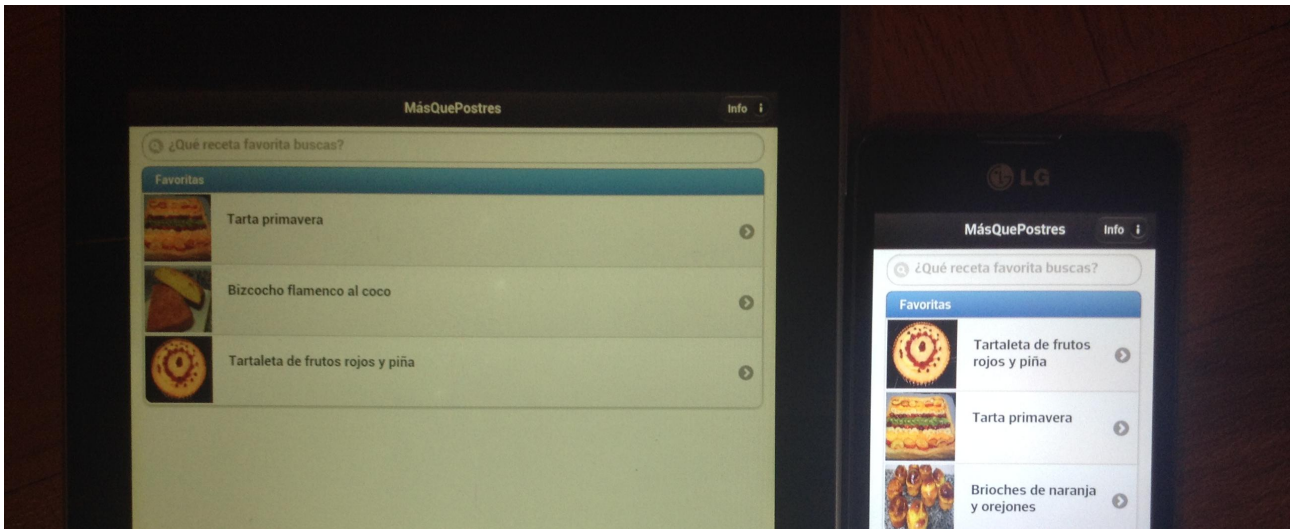
Una vez seleccionada la receta deseada, podremos ver la ficha con los detalles de la misma. Entre ellos hay la opción de **Añadir a favoritas** y añadir **Ingredientes a la lista compra**. Ambos botones aparecen en negro antes de seleccionarlos, pero una vez pulsados se desactivarán y mostrarán en gris flojo para que el usuario sepa que ya se ha tomado acción. De este modo se evita que el usuario añada los ingredientes a la lista de la compra varias veces. En el caso de añadir a favoritas, la verificación de si ya existe o no en el listado de favoritas se hará internamente tras solicitar que se añada, tal como se planteó en su día en los casos de uso.

Se ha comprobado que la navegación desde la pantalla con los detalles de la receta hacia atrás, o cualquier otra opción de navegación como es: **Recetas, Favoritas, Compra, Info** es correcta en los tres dispositivos probados.

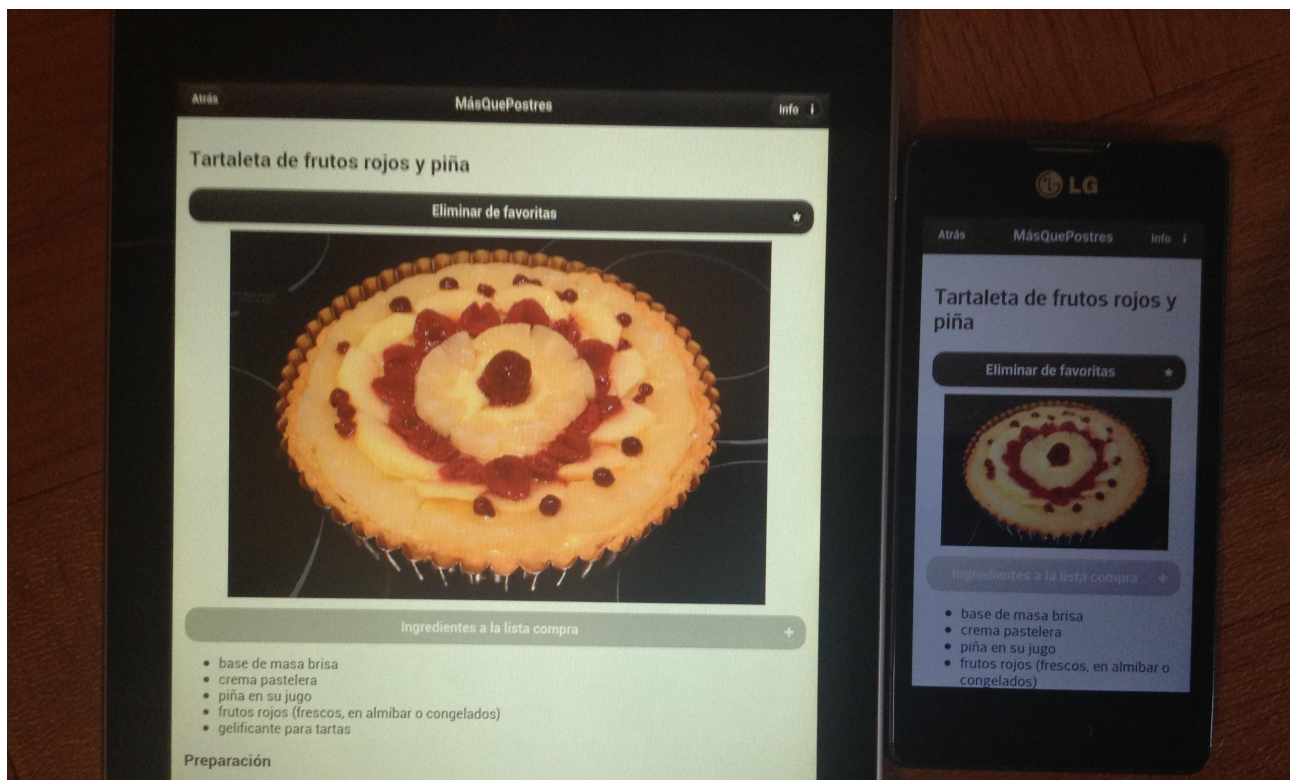




En la opción de **Favoritas** se obtiene el listado de las recetas previamente marcadas como favoritas. También dispone de un filtro de búsqueda en la parte superior.



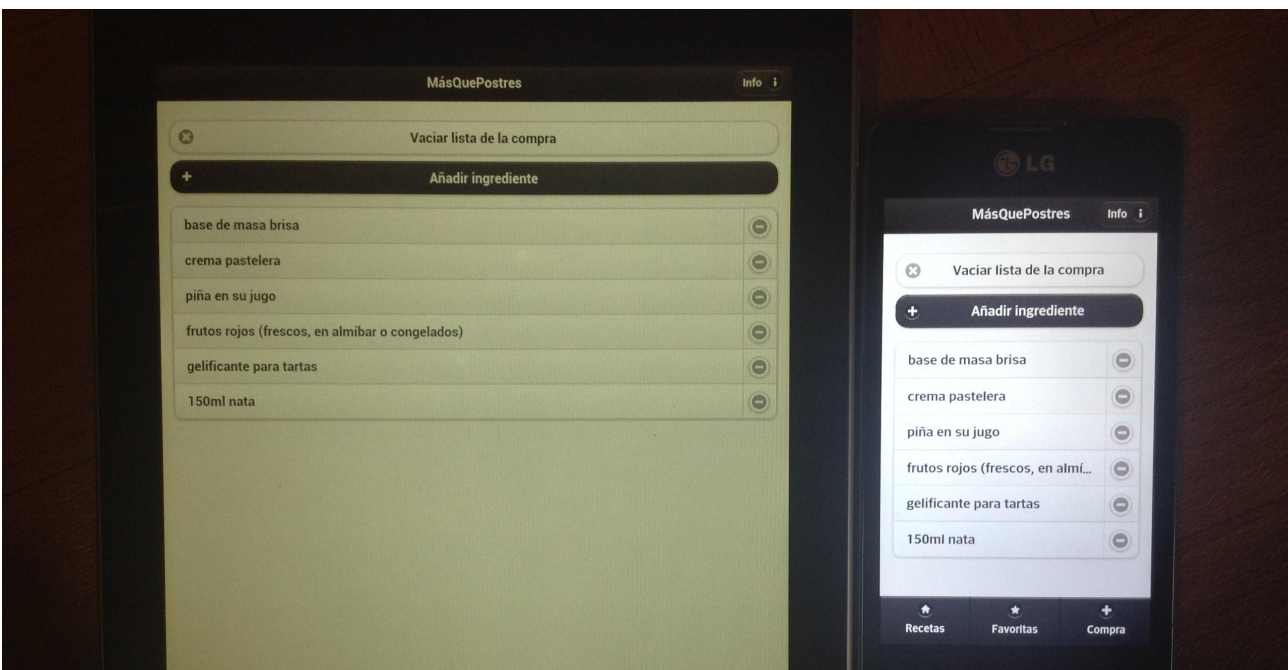
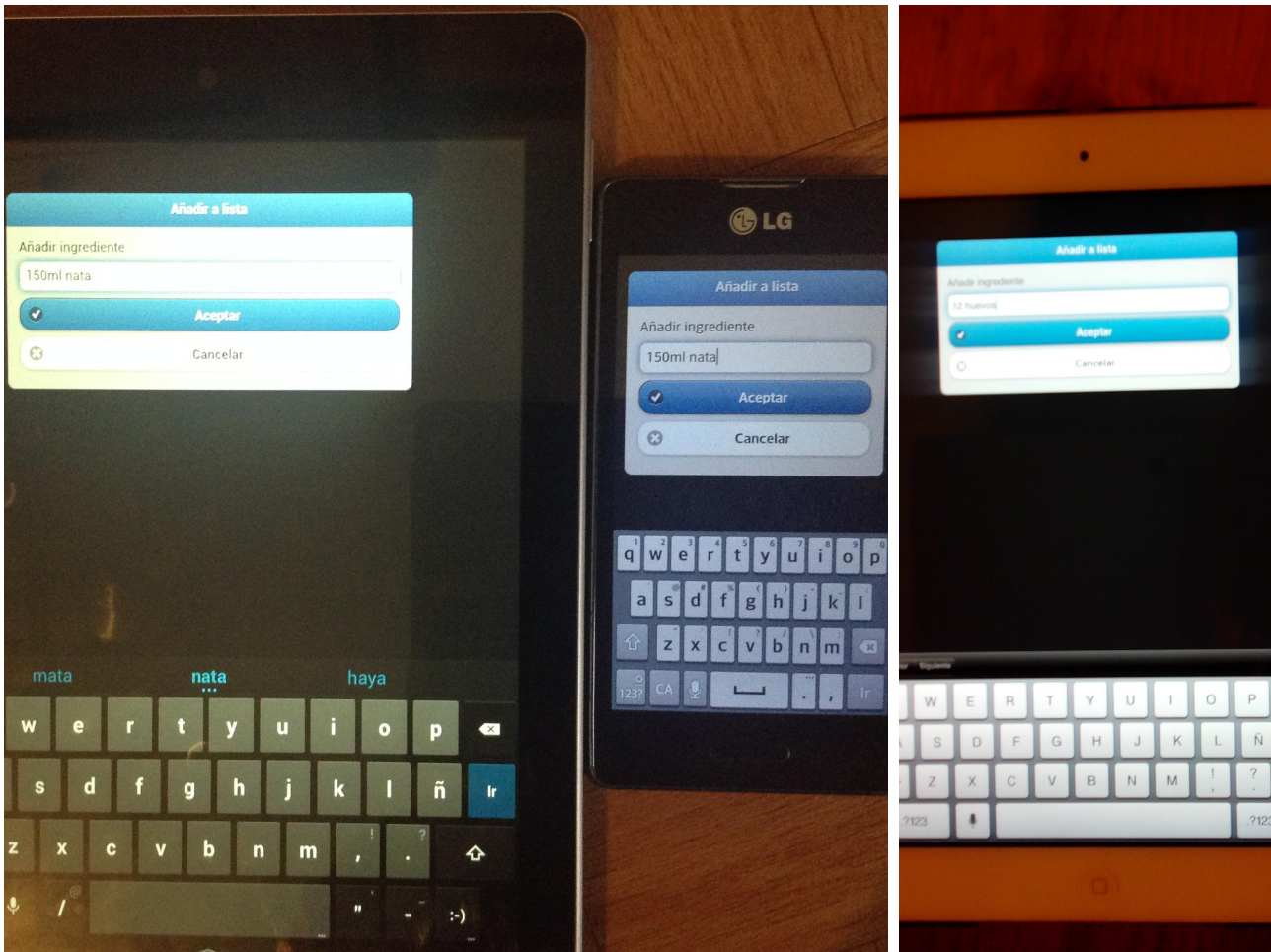
Una vez seleccionada una de ellas se pueden ver los detalles de la misma y se tiene la opción de **Eliminar de favoritas** y añadir **Ingredientes a la lista compra**. En la implementación se ha reutilizado el código empleado para mostrar los detalles de la receta en la opción anterior. Es por esto que el funcionamiento de los botones sigue la misma filosofía. Así no se confunde al usuario.



También se ha comprobado la correcta navegación desde esta pantalla con éxito.

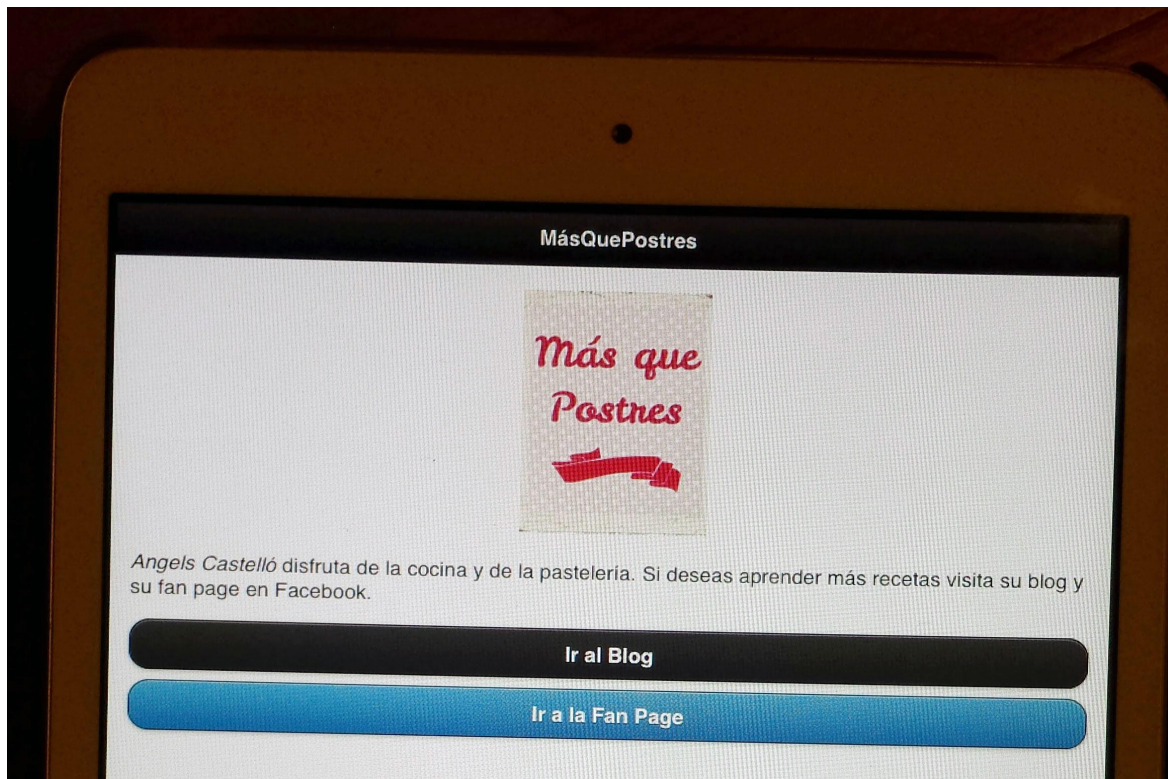


En la pantalla de Compra se pueden gestionar los ingredientes previamente añadidos, es decir, borrar la lista entera, borrar individualmente o crear manualmente un ingrediente.





Se ha comprobado que tanto al pulsar el botón **Aceptar** de la aplicación como el botón **Ir** en el teclado se añade el ingrediente. Al igual, que al cancelar la acción, que se borre el contenido tecleado en el campo de entrada del diálogo.



Por último una muestra de la pantalla de **Info** en el mini iPad.

## 8.4 Parse

Para cargar la base de datos en **Parse** más ágilmente se ha creado un fichero llamado **cargaBD.html** donde se ha incrustado un *script* de JS para cargar la información básica de las recetas excepto la fotografía. Se adjunta un pequeño fragmento para ver la estructura:

```
Parse.initialize(PARSE_APP, PARSE_JS);
var promisesBDLoaded = [];
var RecetaObject = Parse.Object.extend("RecetaObject");
var RecetaCollection = Parse.Collection.extend({
  model: RecetaObject
});

var recetas = new RecetaCollection(
  [{
    "Titulo": "Muffins de arandanos",
    "Ingredientes": ["125g de harina floja", "2 cucharaditas de levadura", "una pizca de sal", "30g de azúcar", "1 huevo", "80ml de leche entera", "30g de mantequilla", "80g de arándanos naturales", "ralladura de medio limón."],
    "Categorias": ["Muffins", "Desayunos"],
    "Preparacion": ["1- Precaentamos el horno a 190°C. Y preparamos los moldes: en el molde de silicona ( o metálico) colocamos una cápsula en cada orificio.", "2- Tamizamos la harina y la mezclamos con la sal y el azúcar.", "3- En otro recipiente batimos el huevo con la leche y la mantequilla derretida y la ralladura de limón. Volcamos esta preparación con los ingredientes secos preparados antes. Mezclamos bien.", "4- No hace falta mezclar demasiado, ya que esta masa debe quedar más o menos grumosa. Añadimos por último los arándanos.", "5- Vertemos la masa en las cápsulas, deben salirnos unos 12, aunque depende del tamaño del mismo. Horneamos a 190°C durante unos 25 min y hasta que al pincharlos el palillo salga limpio. Enfriamos sobre rejilla.", "6- Los muffins no llevan frosting/icing o cobertura, pero al ser un poco seco encuentro que un poco de nata le queda genial, le aporta dulzor y frescor. Así que montamos un poco de nata con azúcar (o azúcar avainillado) y decoramos nuestro muffin."],
  }, {
```

Este es el *dashboard* que nos ofrece Parse para la gestión de los datos. Como muestra, una pantalla donde se ve **CategoriaObject** y otra donde se observa un fragmento de **RecetaObject**.

| objectId   | String | Titulo             | String | createdAt           | Date | updatedAt           | Date | ACL         | ACL | Foto                   | File |
|------------|--------|--------------------|--------|---------------------|------|---------------------|------|-------------|-----|------------------------|------|
| CF6xgKjX   |        | Panes              |        | Nov 06, 2013, 09:49 |      | Nov 06, 2013, 09:49 |      | (undefined) |     | panes_80x80.jpg        |      |
| gkoPzysq1  |        | Las Fáciles        |        | Nov 06, 2013, 09:48 |      | Nov 06, 2013, 10:03 |      | (undefined) |     | lasFaciles_80x80.jpg   |      |
| bVcUnJfUos |        | Galletas           |        | Nov 06, 2013, 09:48 |      | Nov 06, 2013, 09:48 |      | (undefined) |     | galletas_80x80.jpg     |      |
| k2ZCeV7rO  |        | Bollos             |        | Nov 06, 2013, 09:44 |      | Nov 06, 2013, 10:10 |      | (undefined) |     | bolos_80x80.jpg        |      |
| TzqtqEau3  |        | Postres con frutas |        | Oct 24, 2013, 07:44 |      | Nov 06, 2013, 10:07 |      | (undefined) |     | postesFrutas_80x80.jpg |      |
| 7pIEKZK2Zb |        | Pastelitos         |        | Oct 24, 2013, 07:44 |      | Nov 06, 2013, 09:49 |      | (undefined) |     | pastelitos_80x80.jpg   |      |
| Bu4Q7mdLoY |        | Desayunos          |        | Oct 24, 2013, 07:36 |      | Nov 06, 2013, 10:13 |      | (undefined) |     | desayunos_80x80.jpg    |      |
| 3zzeMlUtC  |        | Muffins            |        | Oct 24, 2013, 07:35 |      | Nov 06, 2013, 09:56 |      | (undefined) |     | muffins_80x80.jpg      |      |
| yci5UteGjE |        | Tartas             |        | Oct 23, 2013, 18:32 |      | Nov 06, 2013, 10:00 |      | (undefined) |     | tartas_80x80.jpg       |      |

| objectId   | String | Titulo                             | String | Categorias                              | Array | Ingredientes                                   | Array | Foto                            | File |
|------------|--------|------------------------------------|--------|---|-------|--|-------|---------------------------------|------|
| pKSDxyN8k3 |        | Tarta mousse de fresa (semifrio)   |        | ["Postres con frutas", "Tartas"]        |       | ["plancha de bizcocho", "150g de queso ...     |       | Tarta_mousse_fresa.jpg          |      |
| OZVUMn3SZE |        | Tartaleta de frutos rojos y piña   |        | ["Postres con frutas", "Tartas"]        |       | ["base de masa brisa", "crema pastelera", ...  |       | Tartaleta_frutos_rojos.jpg      |      |
| Bjgo2jdtNj |        | Alfajores de dulce de leche        |        | ["Pastelitos"]                          |       | ["200g de mantequilla", "150g de azúcar", ...  |       | Alfajores.jpg                   |      |
| Pcm0wpBr9p |        | Pastelitos de coco y chocolate ... |        | ["Pastelitos", "Muffins"]               |       | ["175g de coco rallado", "4 cdas de leche, ... |       | Pastelitos_coco_chocoBlanco.jpg |      |
| Y9p075bq9B |        | Muffins de chocolate blanco        |        | ["Bollos", "Muffins"]                   |       | ["150g de chocolate blanco", "1 cucharas, ...  |       | Muffins_chocoBlanco.jpg         |      |
| KdPEei7KMK |        | Espirales vienas de canela         |        | ["Galletas", "Pastelitos"]              |       | ["175g de mantequilla ablandada", "55g ...     |       | Espirales_vienas.jpg            |      |
| 1AMXGMjFtO |        | Muffins rellenas de galletas oreo  |        | ["Desayunos", "Muffins"]                |       | ["300g de harina", "1/2 sobre de levadura, ... |       | Muffins_Oreo.jpg                |      |
| Wh00fQLVpM |        | Scones de chocolate                |        | ["Bollos", "Desayunos"]                 |       | ["115g de mantequilla fría troceada", "45 ...  |       | Scones_choco.jpg                |      |
| 2ZxX6Z1zwL |        | Donuts caseros                     |        | ["Bollos", "Desayunos"]                 |       | ["250g de harina de fuerza", "60g de azú ...   |       | Donuts_caseros.jpg              |      |
| ihelp5PeGK |        | Magdalenas integrales              |        | ["Desayunos", "Las Fáciles", "Muf ...   |       | ["300g de harina integral", "200g de azú ...   |       | Magdalenas_integrales.jpg       |      |
| SP8u0R0Ray |        | Pan de semillas                    |        | ["Desayunos", "Las Fáciles", "Pan ...   |       | ["2 cucharadas de zumo de limón", "300 ...     |       | Pan_semillas.jpg                |      |
| oaq66McXBM |        | Pan de tomates secados al sol      |        | ["Desayunos", "Las Fáciles", "Pan ...   |       | ["1 yema de huevo", "200ml de agua calle ...   |       | Pan_tomates_secados.jpg         |      |
| 4279VJyKjz |        | Tarta de manzanas y almendras      |        | ["Las Fáciles", "Tartas"]               |       | ["300g de pasta quebrada", "10 manzan ...      |       | Tarta_manzanas_almendras.jpg    |      |
| iNVOTjuKJe |        | Magdalenas de cacao                |        | ["Desayunos", "Las Fáciles", "Muf ...   |       | ["150g de huevos enteros", "160g de az ...     |       | Magdalenas_cacao.jpg            |      |
| H3WwJT3K3  |        | Galletas de mantequilla            |        | ["Galletas", "Las Fáciles"]             |       | ["115g de mantequilla", "175g de harina, ...   |       | Galletas_mantequilla.jpg        |      |
| 110TZ0IMmV |        | Galletas de canela                 |        | ["Galletas", "Las Fáciles"]             |       | ["125g de mantequilla ablandada", "350g ...    |       | Galletas_canela.jpg             |      |
| IFAoPtr2hr |        | Tatin de manzana                   |        | ["Las Fáciles", "Postres con fruta ...  |       | ["1kg de manzanas golden (amarillas)", "1 ...  |       | Tatin_manzana.jpg               |      |
| JHBcIxcDV  |        | Bizcocho flamenco al coco          |        | ["Desayunos", "Las Fáciles"]            |       | ["4 huevos", "250g de azúcar", "250g de ...    |       | Bizcocho_flamenco.jpg           |      |
| mz9EEY53wm |        | Tartaletas de plátano y nisperos   |        | ["Las Fáciles", "Pastelitos", "Post ... |       | ["3 plátanos pequeños", "12 nisperos", "9 ...  |       | Tartaletas_platano_nisperos.jpg |      |
| x2R0vkRox7 |        | Galletas 5 semillas                |        | ["Galletas", "Las Fáciles"]             |       | ["250g de harina integral", "100ml de acei ... |       | Galletas_5_semillas.jpg         |      |



## 9 Conclusiones

Personalmente, se está muy satisfecho con el resultado obtenido al final del proyecto. Ha sido un trabajo intenso y laborioso al no tener ningún conocimiento sobre desarrollo de aplicaciones móviles al inicio del proyecto. Por ello ha sido doblemente satisfactorio todo el proceso, desarrollar la aplicación y obtener un gran bagaje de nuevos conocimientos.

Como punto negativo, podemos concluir que jQuery Mobile es idóneo para desarrollar aplicaciones móviles sencillas. Pero si se desea un diseño más personal y una aplicación de mayor complejidad no es lo más adecuado su uso.

### 9.1 Logro de objetivos

Se puede concluir que ha sido un éxito todo el proyecto ya que se han logrado, tras un duro trabajo, todos los objetivos marcados desde el inicio del mismo:

- ✓ Aplicación multiplataforma con un único desarrollo en HTML5, CSS3, JavaScript y jQueryMobile. Sin necesidad de aprender los lenguajes específicos de cada plataforma gracias a PhoneGap Build.
- ✓ Aplicación compatible en diferentes dispositivos móviles. Este objetivo era de los más difíciles dada la gran cantidad de resoluciones de pantalla existentes, en concreto en terminales Android. Se ha probado con éxito en tres dispositivos con diferente tamaño de pantalla y sistema operativo. Concretamente con iOS 7 y Android 4.x.x.

Adicionalmente, durante la grabación del video para la memoria se ha probado la aplicación en un cuarto dispositivo: Samsung Galaxy S4 con Android 4.3. El resultado ha sido completamente satisfactorio.

- ✓ Aprendizaje y uso de las diferentes técnicas de acceso y almacenamiento de información: dispositivo local y servidor remoto.
- ✓ Utilización de un servicio básico como *backend*. Esto ha implicado un aprendizaje de nuevas tecnologías muy interesante como es el uso de Parse como *BaaS*.
- ✓ Desarrolladas en la aplicación todas las funcionalidades definidas en el proyecto.
- ✓ Aplicación final con cambios mínimos respecto al diseño inicial previsto en la fase de diseño. Se trata de una aplicación que radica su éxito en su sencillez y funcionalidad.

### 9.2 Líneas futuras

Gracias al valioso aporte de Jordi Ceballos, se ha detectado que la aplicación cuando no se usa una conexión wifi de calidad es en algunos momentos lenta y tarda un poco en cargar, por lo que el principal objetivo a trabajar será optimizar el rendimiento de la aplicación. Este problema

no había sido detectado durante todo el proceso de implementación, dado que siempre se trabajó con una conexión wifi de calidad y su rendimiento fue excelente.

Otra línea futura de trabajo sería cambiar el planteamiento de los ingredientes como un simple *string* y diseñar una nueva base de datos donde para los ingredientes se haga distinción entre cantidades y tipo de ingrediente. De este modo se podría desarrollar nuevas funcionalidades más prácticas:

- ✓ Sumar y mostrar cantidades totales de un mismo ingrediente en la lista de la compra.
- ✓ Permitir en un futuro, añadir la información de para cuantas personas es la receta y a su vez poder enviar a la lista de la compra los ingredientes necesarios para las “X” personas que deseemos. La aplicación realizaría los cálculos necesarios.

Durante la implementación de la aplicación han surgido nuevas ideas para mejorar la aplicación y así obtener un producto de mayor calidad antes de publicarla en las diferentes tiendas de aplicaciones:

- ✓ Registrar usuario a través de su cuenta de Facebook y uso de Facebook JavaScript SDK en la aplicación para poder gestionar otras nuevas funcionalidades.
- ✓ Añadir funcionalidades: votación recetas, recomendar recetas, etc.
- ✓ Publicar la aplicación en las tiendas de distribución de aplicaciones como son: App Store y Google Play.
- ✓ Investigar posible compatibilidad / adaptación de la aplicación en dispositivos móviles con sistema operativo Firefox OS.

## 10 Instalación de la aplicación

A pesar de estar desarrollada la aplicación para móviles iOS, todavía no se puede instalar en ellos ya que aún no se ha obtenido el certificado para producción. Actualmente sólo se ha obtenido el certificado para desarrollo.

Se puede instalar la aplicación en cualquier dispositivo móvil Android 4.x.x aunque no esté publicada en ninguna tienda de aplicaciones. Hay dos modos de instalación:

- 1) Instalar paquete .APK tras descargarlo del siguiente enlace público de Google Drive <https://docs.google.com/file/d/0B3ck-P6iHkRyRG0wYlluNjBpYVvk/edit> o desde esta dirección URL más corta que se ha creado para facilitar su acceso <http://tinyurl.com/p5aygfc>
- 2) Leyendo con un lector de códigos QR como puede ser QR Droid, el siguiente código QR generado por PhoneGap Build

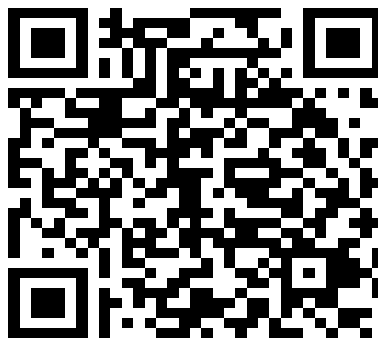


Figura 17 – Código QR de la aplicación

## 11 Fuentes de información

### 11.1 Bibliografía

CHETAN, K JAIN (2012). *jQuery Mobile Cookbook*. UK: Packt Publishing

GIFFORD, MATT (2012). *PhoneGap Mobile Application Development Cookbook*. UK: Packt Publishing

### 11.2 Recursos web

Christophe Coenraets blog [en línea]

<http://coenraets.org/blog/>

Codiqa [en línea]

<https://codiqa.com/>

jQuery Mobile 1.2 Demos [en línea]

<http://jquerymobile.com/demos/1.2.1/>

jQuery Mobile 1.3 API Documentation [en línea]

<http://api.jquerymobile.com/>

Parse JavaScript Guide [en línea]

[https://www.parse.com/docs/js\\_guide#javascript\\_guide](https://www.parse.com/docs/js_guide#javascript_guide)

PhoneGap Build [en línea]

<https://build.phonegap.com/apps>

Raymond Camden's blog [en línea]

<http://www.raymondcamden.com/>

## 12 Índice de figuras

- Figura 1 – Fases del proyecto
- Figura 2 – Planificación del proyecto
- Figura 3 – Diagrama casos de uso
- Figura 4 – Patrón MVC (Model, View, Controller)
- Figura 5 – Diagrama de clases global
- Figura 6 – Pantalla Home
- Figura 7 – Pantalla RecetasPage
- Figura 8 – Pantalla RecetasFicha
- Figura 9 – Pantalla RecetasFicha
- Figura 10 – Pantalla FavoritasPage
- Figura 11 – Pantalla FavoritasFicha
- Figura 12 – Pantalla FavoritasFicha
- Figura 13 – Pantalla ComprasPage
- Figura 14 – Pantalla DialogoIngrediente
- Figura 15 – Pantalla InfoPage
- Figura 16 – Flujo de navegación entre pantallas
- Figura 17 – Código QR de la aplicación